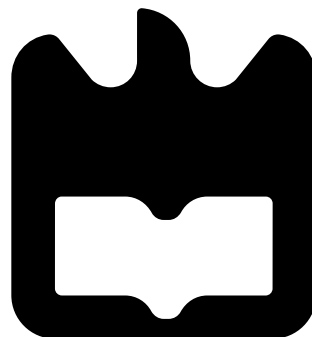




**Ana Carolina
Santos Cruz
Aspeçada Martins**

**Mecanismos de Disseminação de Mensagens de
Emergência numa Rede de Veículos**





**Ana Carolina
Santos Cruz
Aspeçada Martins**

Mecanismos de Disseminação de Mensagens de Emergência numa Rede de Veículos

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica da Professora Doutora Susana Sargento, Professora Associada com Agregação do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e co-orientação científica do Doutor Carlos Senna, Investigador do Instituto de Telecomunicações de Aveiro.

o júri / the jury

presidente / president

Professor Doutor Manuel Bernardo Salvador Cunha

Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro (por delegação da Reitora da Universidade de Aveiro)

vogais / examiners committee

Professora Doutora Susana Isabel Barreto de Miranda Sargento

Professora Associada com Agregação do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro (orientadora)

Professora Doutora Marília Pascoal Curado

Professora Auxiliar com Agregação do Departamento de Engenharia Informática da Universidade de Coimbra

agradecimentos

Após o término desta dissertação, tenho a realçar que, por mais baixos que existam ao longo do percurso académico, existem sempre os altos que nos ajudam a encontrar um rumo.

Quero, então agradecer e dedicar a conclusão deste percurso aos meus pais, irmão e avós. Aos meus pais e irmão agradeço por terem acreditado em mim e me apoiarem nos bons e menos bons momentos, ao longo do curso. Obrigado por toda a paciência e confiança em mim, mesmo quando não acreditava ser possível vocês estiveram lá para me apoiar como sempre. Ao meu avô agradeço toda a paciência e compreensão ao longo deste percurso, mesmo que nem sempre percebesse muito bem esta longa vida de estudante universitária. Sei que estás contente por saber que cheguei ao fim.

Quero agradecer ao meu namorado Sérgio Manso por sempre acreditar em mim, por todo o amor, apoio e dedicação constante durante estes anos. Obrigada por estares sempre do meu lado e por nos mantermos unidos, mesmo nas situações mais difíceis por que já passámos. És muito importante e sei que posso confiar em ti e contar contigo para os bons e maus momentos. Obrigada por tudo e pela enorme paciência que tiveste nestes últimos meses!

Agradeço à Susana Lopes por ser uma amiga de longa data (quase irmã), que apesar de distante está sempre presente nos momentos bons e nos mais difíceis. És a prova que a amizade supera todos os obstáculos que vão surgindo nas nossas vidas. Obrigada por acreditares em mim e me ajudares mesmo estando longe!

Por fim, mas não menos importante quero também agradecer à Deolinda Moura por todas as horas e minutos de paciência, risadas e companheirismo ao longo dos últimos anos e sobretudo deste último. Foste e és sem dúvida uma grande amiga que espero levar para a vida.

Quero agradecer à professora Susana Sargento e ao Carlos Senna pela orientação ao longo deste período e também pela oportunidade de poder fazer dissertação numa área que gosto e que considero ser interessante, atual e em constante evolução. As atividades extra trabalho e os jantares de convívio sempre animados, foram iniciativas que contribuem para conhecer melhor os colegas e integrar sucessivamente os novos colegas que vão chegando ao NAP.

Ao grupo NAP agradeço todos os momentos bem passados, as brincadeiras e boa disposição. Quero deixar um agradecimento, em particular, à fila de trás do laboratório, por todas as horas de brincadeiras, risadas e momentos bem passados que levo comigo. Obrigado a todos vocês! Quero deixar um particular agradecimento ao Carlos Ferreira por toda a paciência, disponibilidade e prontidão em ajudar, assim como a constante boa disposição.

"Never stop fighting until the fight is done!"

The Untouchables

Palavras-Chave

VANET, Mensagens de Emergência, V2V, IEEE 802.11p, WAVE, Zona de Relevância, Protocolos Geocast

Resumo

Nas últimas décadas, a tecnologia e as telecomunicações têm sofrido uma evolução exponencial. Atualmente, cada vez mais as comunicações entre todo o tipo de veículos assumem um papel importante no quotidiano de cada um de nós, quer seja pela necessidade de obter informações antecipadas acerca do percurso rodoviário que fazemos diariamente, ou até mesmo a distribuição de conteúdos urgentes e não urgentes. Deste modo, as redes veiculares surgem com algumas particularidades que as distinguem de outro tipo de redes, tais como o tempo reduzido de contacto que leva à perda de ligações e conectividade, devido à mobilidade constante dos nós na rede, a sua dispersão geográfica e a densidade variável. Estas singularidades criam algumas fragilidades na rede e proporcionaram o aparecimento de um novo conceito denominado Redes Tolerantes a Atrasos (*Delay Tolerant Network (DTN)*). Desta forma, o encaminhamento da informação é feito através de um mecanismo de *Store-Carry-and-Forward (SCF)*, que resolve os desafios descritos anteriormente, permitindo que a informação chegue a todos os nós em situações onde não existe um caminho estabelecido *end-to-end*.

Uma vez que o trabalho desenvolvido tem por base uma rede veicular *ad-hoc* e o objetivo principal é a disseminação de conteúdo urgente nessa rede veicular, foi necessário tirar partido dos mecanismos da plataforma DTN. Desta forma, foi possível garantir que o conteúdos chegassem a todos os veículos que se encontravam numa área delimitada pelas coordenadas *Global Position System (GPS)* do acidente, em particular, atrás do mesmo e na mesma direcção do movimento, recorrendo ao número mínimo de saltos possíveis. Outro objetivo foi minimizar o número de mensagens repetidas na área de interesse. A mensagem encaminhada foi gerada pelo veículo que teve o acidente ou pelo veículo que deteta o acidente (através de uma travagem brusca, por exemplo).

De forma a tirar partido da localização ou do número de vizinhos que cada veículo tem num dado instante temporal, foram testados e implementados três algoritmos: um para alta densidade, outro para baixa e outro que serve de comparação com os dois anteriores. Estas estratégias foram ser implementadas na plataforma *mobile Opportunistic VEhicular Emulator for Real Scenarios (mOVERS)*, onde também foi avaliado o desempenho das mesmas, por forma a ser escolhida a que apresenta melhor desempenho, tendo em conta, a entrega de pacotes úteis e a quantidade de pacotes redundantes na rede.

Concluiu-se que a melhor estratégia foi a de Alta Densidade para os cenários estudados, uma vez que a taxa de pacotes úteis entregues era muito elevada e a taxa de pacotes repetidos, apesar de não ser tão baixa quanto o desejável, era bastante inferior à da estratégia de *Broadcast*.

Keywords

VANET, Emergency Messages, V2V, IEEE 802.11p, WAVE, Zone of Relevance, Geocast Protocols

Abstract

In last decades, the technology and telecommunications have suffered an exponential evolution. Nowadays, the communications between all types of vehicles have an important role in daily life of each of us, whether it is the need to know information in advance about the vehicles route, or even the distribution of urgent or non-urgent content. In this way, the vehicular networks have special characteristics that distinguish them from other types of networks, such as the reduced contact time that leads to loss of connections and connectivity, due to node mobility in the network, geographic dispersion and variable density. Those singularities create a fragility in the network and have led to the emergence of a new concept called DTN. Thus, the forwarding of the information is done by a mechanism called SCF, which addresses the challenges described before, allowing information to reach all nodes in situations where there is no established end-to-end path.

Since the work developed is based on an ad-hoc network and the main objective is the dissemination of urgent content in this network, it was necessary to take advantage of the mechanisms of the DTN. Thus, it is possible to ensure that it reaches all the vehicles which are in a delimited area by GPS coordinates of the accident. In particular, behind the accident and in the same direction as the movement, using the minimum number of hops. Another objective is to minimize the number of repeated messages in the relevant area. The message that will be forwarded is generated by the vehicle which had an accident or by the vehicle that detected the accident (through a strong break, as example).

To take advantage of location or number of neighbors that each vehicle has in a temporal instant, it was tested and implemented three strategies: one for high density, another for low density and the other for comparison with the previous two. These strategies will be implemented in a platform called mOVERS, which will also evaluate the performance, in order to choose the one that presents best performance, taking into account delivery of useful packets and amount of redundant packets in the network.

It can be concluded that the best strategy is the one of High Density, since the rate of useful packets delivered is very high and the rate of repeated packets, although not as low as desirable, is much lower than that of the strategy of Broadcast.

Conteúdo

Conteúdo	i
Lista de Figuras	iii
Lista de Tabelas	v
Lista de Algoritmos	vii
Lista de Acrónimos	ix
1 Introdução	1
1.1 Contexto e Motivação	1
1.2 Objetivos e Contribuições	2
1.3 Organização da Dissertação	3
2 Estado de Arte	5
2.1 Introdução	5
2.2 Vehicular Ad hoc NETworks (VANETs)	5
2.2.1 Disseminação de Dados	6
2.2.2 Pilha de Protocolos para VANETs	7
2.2.2.1 Institute of Electrical and Electronics Engineers (IEEE) 802.11p	8
2.2.2.2 IEEE 1609.x	9
2.2.3 Aplicações e Serviços	9
2.2.4 Desafios	10
2.3 Protocolos de Encaminhamento aplicados a VANETs	11
2.3.1 Protocolos <i>Broadcast</i>	12
2.3.2 Protocolos <i>Geocast</i>	12
2.4 Delay Tolerant Networks (DTNs)	15
2.4.1 Definição	15
2.4.2 Arquitetura	16
2.4.3 Mecanismo de SCF	17
2.5 Trabalhos Relacionados baseados em <i>clusters</i>	18
2.6 Considerações do Capítulo	21
3 Estratégias para a Disseminação de Mensagens de Emergência	23
3.1 Descrição do Capítulo	23
3.2 Problema	23

3.3	Disseminação de Conteúdos Urgentes - Solução Proposta	25
3.3.1	Estratégia de Alta Densidade	26
3.3.2	Estratégia de Baixa Densidade	31
3.3.3	Estratégia de <i>Broadcast</i>	35
3.4	Considerações do Capítulo	37
4	Implementação e Integração	39
4.1	Descrição do Capítulo	39
4.2	Descrição do mOVERS	39
4.2.1	Visão Geral da Arquitetura e Funcionamento	39
4.2.2	Recolha de Dados da Base de Dados MySQL	41
4.2.3	Descrição dos módulos do mOVERS	42
4.3	Integração e Implementação das Estratégias de disseminação de mensagens de emergência no mOVERS	46
4.3.1	Modificações Gerais	46
4.3.2	Restrições do mOVERS para a disseminação de mensagens de emergência	52
4.4	Considerações do Capítulo	53
5	Testes e Resultados	55
5.1	Descrição do Capítulo	55
5.2	<i>Software</i> e Equipamento	55
5.3	<i>Scripts</i>	56
5.4	Descrição do Cenário	56
5.5	Avaliação do Desempenho das Estratégias	57
5.5.1	Zona de Relevância de 3Km	57
5.5.2	Zona de Relevância de 5Km	60
5.5.3	Zona de Relevância de 7Km	63
5.6	Considerações do Capítulo	68
6	Conclusões e Trabalho Futuro	69
6.1	Conclusões	69
6.2	Trabalho Futuro	70
	Bibliografia	71

Lista de Figuras

1.1	Aplicações em cenários de emergência	2
2.1	Arquitetura de uma VANET	6
2.2	Canais e Espectro da Largura de Banda do Dedicated Short Range Communications (DSRC) de acordo com a norma Europeia IEEE 802.11p	8
2.3	Protocolo Wireless Access in Vehicular Environments (WAVE)	8
2.4	Classificação dos protocolos de encaminhamento em VANETs	12
2.5	Processo de descoberta de rotas usando mensagens Zone Route REQuest (ZRREQ) e Zone Route REply (ZRREP)	13
2.6	Cenário de estrada usando o método baseado na distância	14
2.7	Processo de seleção dos nós retransmissores usando o protocolo Inter-Vehicles Geocast (IVG)	16
2.8	Protocolo DTN	16
2.9	Arquitetura de Encaminhamento de um <i>Bundle</i>	17
2.10	Mecanismo SCF	18
2.11	Arquitetura de <i>broadcast</i> de mensagem de emergência baseada em <i>clusters</i> para VANETs	19
2.12	Arquitetura do modelo de auto-estrada baseada em <i>clusters</i> para VANETs	20
2.13	Arquitetura Vehicular Backbone Network (VBN) aplicada a VANETs	21
3.1	Cenário de estudo	26
3.2	Cenário da estratégia de Alta Densidade	27
3.3	Diagramas de fluxo das lógicas dos pacotes de emergência correspondentes da estratégia de Alta Densidade	28
3.4	Direção da Disseminação do movimento	30
3.5	Cenário da estratégia de Baixa Densidade	32
3.6	Diagramas de fluxo das lógicas dos pacotes de emergência correspondentes à estratégia de Baixa Densidade	33
3.7	Diagramas de fluxo das lógicas dos pacotes de <i>advertisements</i> correspondentes à estratégia de Baixa Densidade	35
3.8	Diagramas de fluxo das lógicas dos pacotes de emergência correspondentes à estratégia de <i>Broadcast</i>	37
4.1	Arquitetura do mOVERS	40
4.2	Classe do cabeçalho das mensagens do mOVERS	41
4.3	Classe das mensagens de <i>advertisement</i> do mOVERS	42
4.4	Classes dos vários tipos de Routing	43

4.5	Classes dos vários tipos de vizinhos	45
4.6	Organização do módulo Storage	46
4.7	Arquitetura do mOVERS - modificações	46
4.8	Estrutura do Pacote do mOVERS - modificações	48
4.9	Classes de <i>Routing</i> do mOVERS	49
4.10	Módulo de <i>Logging</i> alterado	50
4.11	Fluxograma do Funcionamento do Módulo de <i>Logging</i>	51
5.1	Avaliação do desempenho para a Zone of Relevance (ZoR) de 3 Km e WAVE de 500m: número de iterações	58
5.2	Avaliação do desempenho para a ZoR de 3 Km e WAVE de 500m: taxa de entrega	58
5.3	Avaliação do desempenho para a ZoR de 3 Km e WAVE de 500m: taxa de pacotes repetidos	59
5.4	Avaliação do desempenho para a ZoR de 3 Km e WAVE de 700m: número de iterações	59
5.5	Avaliação do desempenho para a ZoR de 3 Km e WAVE de 700m: taxa de entrega	60
5.6	Avaliação do desempenho para a ZoR de 3 Km e WAVE de 700m: taxa de pacotes repetidos	60
5.7	Avaliação do desempenho para a ZoR de 5 Km e WAVE de 500m: número de iterações	61
5.8	Avaliação do desempenho para a ZoR de 5 Km e WAVE de 500m: taxa de entrega	61
5.9	Avaliação do desempenho para a ZoR de 5 Km e WAVE de 500m: taxa de pacotes repetidos	62
5.10	Avaliação do desempenho para a ZoR de 5 Km e WAVE de 700m: número de iterações	63
5.11	Avaliação do desempenho para a ZoR de 5 Km e WAVE de 700m: taxa de entrega	63
5.12	Avaliação do desempenho para a ZoR de 5 Km e WAVE de 700m: taxa de pacotes repetidos	64
5.13	Avaliação do desempenho para a ZoR de 7 Km e WAVE de 500m: número de iterações	65
5.14	Avaliação do desempenho para a ZoR de 7 Km e WAVE de 500m: taxa de entrega	65
5.15	Avaliação do desempenho para a ZoR de 7 Km e WAVE de 500m: taxa de pacotes repetidos	66
5.16	Avaliação do desempenho para a ZoR de 7 Km e WAVE de 700m: número de iterações	66
5.17	Avaliação do desempenho para a ZoR de 7 Km e WAVE de 700m: taxa de entrega	67
5.18	Avaliação do desempenho para a ZoR de 7 Km e WAVE de 700m: taxa de pacotes repetidos	67

Lista de Tabelas

3.1	Características dos vários protocolos	24
4.1	Base de Dados do mOVERS : Tabela por <i>Timestamps</i>	41
4.2	Base de Dados do mOVERS : Tabela por Vizinhos	41
4.3	Base de Dados do mOVERS : Tabela por Road Side Units (RSUs)	42

Lista de Algoritmos

1	Cálculo do <i>MaxRank</i> para a estratégia de Alta Densidade	29
2	Lógica de envio de <i>advertisements</i> para estratégia de Alta Densidade	31
3	Lógica de receção de <i>advertisements</i> para estratégia de Alta Densidade	31
4	Cálculo do <i>MaxRank</i> para a estratégia de Baixa Densidade	34

Lista de Acrónimos

ADV	Advertisement
AODV	Ad hoc On-Demand Distance Vector
API	Application Programming Interface
BSS	Basic Service Set
CBL	Cluster Black Lists
CC	Canal de Controlo
CF	Cluster Forwarder
CFL	Cluster Forwarder Lists
CH	Cluster Head
CLA	Convergence Layer Adapter
CPU	Central Processing Unit
CTB	Clear To Broadcast
CS	Canal de Serviço
DRG	Distributed Robust Geocast
DTN	Delay Tolerant Network
DSRC	Dedicated Short Range Communications
DSCF	Directional Store-Carry-and-Forward
ESPM	Emergency Situation Prediction Mechanism
ETSI	European Telecommunications Standards Institute
FCC	Federal Communication Commission
FFRDV	Fastest Ferry Routing in DTN-enabled Vehicular Ad-hoc
GPS	Global Position System
ID	Identificador

IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPC	Inter-Process Communication
IT	Instituto de Telecomunicações
ITS	Intelligent Transportation Systems
IVG	Inter-Vehicles Geocast
MAC	Medium Access Control
mOVERS	mobile Opportunistic VEhicular Emulator for Real Scenarios
NAP	Network Architecture and Protocols
OBU	On Board Unit
OSI	Open Systems Interconnection
RAM	Random-Access Memory
RAP	Road Accident Prevention
ROVER	RObust VEhicular Routing
RTB	Request To Broadcast
RSU	Road Side Unit
RSSI	Received Signal Strength Indicator
SCF	Store-Carry-and-Forward
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VANET	Vehicular Ad hoc NETwork
VBN	Vehicular Backbone Network
VDTN	Vehicular Delay Tolerant Network
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
WAVE	Wireless Access in Vehicular Environments
Wi-Fi	Wireless Fidelity
WME	WAVE Management Entity
WSMP	WAVE Short Message Protocol

ZFR	Zona Fora de Risco
ZoR	Zone of Relevance
ZoF	Zone of Forwarding
ZMQ	Zero-M Queue
ZRE	Zona de Risco Elevado
ZRM	Zona de Risco Médio
ZRB	Zona de Risco Baixo
ZRREQ	Zone Route REQuest
ZRREP	Zone Route REPlY

Capítulo 1

Introdução

As *Vehicular Ad hoc NETWORKs (VANETs)* oferecem uma variedade de serviços, entre os quais a possibilidade de disseminar mensagens pelos veículos. Estratégias eficientes para a disseminação de mensagens em situações de emergência têm requisitos especiais. Esta secção discute o contexto e a motivação para construir essas estratégias e mostra como, no âmbito desta dissertação, foram elaboradas algumas propostas para a disseminação de mensagens de emergência em VANETs.

1.1 Contexto e Motivação

As VANETs apresentam características especiais como: a mudança muito rápida da topologia da rede e ligações intermitentes, devido ao tempo de contacto entre os nós ser bastante reduzido [1]. Para lidar com esses desafios que a VANET tem é necessário criar mecanismos que sejam capazes de tolerar as ligações intermitentes, bem como os atrasos longos. Uma arquitetura que é capaz de transmitir informação confiável, tendo em conta os atrasos e as conexões intermitentes designa-se DTN [2]. As Redes Tolerantes a Atrasos (*Delay Tolerant Networks* - DTN) foram, inicialmente, desenvolvidas para permitir comunicações a longas distâncias (entre satélites, por exemplo) e com atrasos longos. Tais características tornam as DTNs uma boa alternativa para lidar com os requisitos das VANETs e, por isso, quando é aplicada às mesmas, designa-se *Vehicular Delay Tolerant Network (VDTN)*.

Desta forma, para a entrega de mensagens urgentes, em especial as mensagens cujo conteúdo sinaliza um estado de emergência que requer um atendimento rápido, é necessário o desenvolvimento de mecanismos que consigam suportar tais requisitos, uma vez que esses mecanismos não são encontrados em VANETs. A distribuição de conteúdos urgentes permite agir em situações de emergência como notificação de acidentes, congestionamento do trânsito, ambulâncias, ultrapassagens, estado da via, entre outros (Figura 1.1).

Assim, o propósito desta dissertação é a disseminação de conteúdos urgentes numa VANET, usando uma plataforma que permite receber as mensagens, armazenar caso não existam veículos no alcance, e disseminá-las de seguida. Para tal, foram implementadas três estratégias (Alta Densidade, Baixa Densidade e *Broadcast*) tendo em conta: a utilização apenas de comunicações *Vehicle to Vehicle (V2V)*, criação de processos de seleção de nós (veículos) que permitam reduzir a quantidade de pacotes repetidos na rede e tirar partido da localização geográfica de cada nó, delimitando uma área, *Zone of Relevance (ZoR)*, onde a mensagem vai ser disseminada. As duas estratégias de Alta e Baixa Densidade, fazem uma seleção de nós,

tendo em conta critérios distintos e foram desenvolvidas para densidades de veículos diferentes (baixa e elevada quantidade de veículos). A terceira (*Broadcast*) não contém qualquer tipo de seleção de nós, nem tem em conta a densidade de veículos, isto é, todos os nós dentro da ZoR enviam a mensagem para todos os seus vizinhos. A plataforma utilizada para implementar e testar as estratégias propostas denomina-se mOVERS e foi desenvolvida em parceria com a Veniam e o grupo investigação Network Architecture and Protocols (NAP) do Instituto de Telecomunicações (IT) da Universidade de Aveiro. O mOVERS foi escolhido por já estar preparado para redes veiculares e já conter os módulos necessários ao desenvolvimento das estratégias de disseminação de mensagens.



Figura 1.1: Aplicações em cenários de emergência [3]

1.2 Objetivos e Contribuições

O objetivo desta dissertação é introduzir mecanismos de envio e encaminhamento de informação, que permitam disseminar diretamente a informação entre os veículos que estão na rede, sem recorrer à comunicação com a infraestrutura. Por exemplo, caso ocorra um acidente ou estejam a decorrer obras na estrada, os veículos irão disseminar essas mensagens apenas entre eles, sem envolverem a infraestrutura. Apesar da intermitência das ligações da rede, a mensagem chega a todos os veículos, pois o mecanismo utilizado possui um mecanismo que permite guardar a mesma até encontrar uma ligação confiável através da qual a informação pode ser encaminhada. No entanto, se não existir uma ou mais estratégias definidas para o encaminhamento da informação urgente, a informação não chegará aos veículos pretendidos ou poderá circular uma elevada quantidade de mensagens repetidas na rede desnecessariamente. De forma a ser possível concretizar esta proposta, as seguintes etapas foram desenvolvidas:

- Estudar os mecanismos existentes para a disseminação de mensagens de emergência e

como adaptar a arquitetura da rede a estes mecanismos;

- Desenvolver mecanismos que disseminem a informação tirando partido da localização geográfica dos veículos (estratégias de disseminação de disseminação de mensagens de emergência);
- Reduzir a quantidade de pacotes redundantes na rede, recorrendo a uma seleção estratégica de veículos que têm como função enviarem as mensagens;
- Testar e avaliar o desempenho dos mecanismos desenvolvidos para a disseminação da informação usando comunicação V2V;

As contribuições desta dissertação são as seguintes:

- Utilização exclusiva da comunicação V2V na rede, de forma a disseminar as mensagens por todos os veículos contidos numa dada área;
- Redução do número de saltos que a informação necessita para chegar da fonte ao destino;
- Desenvolvimento de mecanismos de disseminação de conteúdos urgentes que permitem melhorar o encaminhamento da informação numa situação concreta de emergência, a ocorrência de um acidente.

1.3 Organização da Dissertação

Esta dissertação está organizada em seis capítulos:

- Capítulo 1 - Introdução - Este capítulo descreve o contexto e a motivação desta dissertação, bem como os objetivos, contribuições e a estrutura da mesma.
- Capítulo 2 - Estado de Arte - Este capítulo fornece uma breve descrição do conceito e arquitetura de uma VANET, bem como as aplicações e serviços associados à mesma. Faz uma breve abordagem sobre a tecnologia WAVE e a família IEEE 1609.x, uma vez que é a tecnologia que vai ser considerada ao longo deste trabalho para a comunicação V2V. Descreve também o conceito e arquitetura de uma DTN e o mecanismo de Store-Carry-and-Forward (SCF). Por fim, são apresentados alguns exemplos de protocolos de *Geocast* que são aplicados às mensagens de emergência numa VANET, bem como alguns trabalhos relacionados com disseminação de mensagens de emergência.
- Capítulo 3 - Estratégias para a Disseminação de Mensagens de Emergência - Este capítulo descreve o problema e as soluções propostas (Estratégias de Alta e Baixa Densidade e *Broadcast*).
- Capítulo 4 - Implementação e Integração - Este capítulo apresenta a implementação e integração das três estratégias propostas na plataforma mOVERS, bem como o funcionamento do mOVERS.
- Capítulo 5 - Testes e Resultados - Este capítulo contém uma breve descrição do software usado para os testes, bem como os *scripts* usados para a realização dos gráficos e a avaliação e comparação do desempenho das três estratégias implementadas.

- Capítulo 6 - Conclusões e Trabalho Futuro - Este capítulo contém as conclusões relativamente às soluções desenvolvidas e implementadas no mOVERS, e ainda alguns tópicos sobre trabalho que poderá ser desenvolvido no futuro para melhorar e complementar esta dissertação.

Capítulo 2

Estado de Arte

2.1 Introdução

O foco desta dissertação envolve conceitos e tecnologias relacionadas com *Delay Tolerant Networks (DTNs)*, em particular, as VANETs. Além disso, aspetos relacionados com a disseminação de dados são fundamentais para o entendimento do problema de disseminação de mensagens de emergência tratado nesta dissertação. Desta forma, a seguir é apresentado o estado de arte que contém informações acerca de DTNs, VANETs e protocolos de encaminhamento para VANETs.

2.2 Vehicular Ad hoc NETWORKS (VANETs)

Uma rede veicular é formada por um conjunto de veículos que podem, ou não, estar em movimento. Estes veículos estão equipados com uma *On Board Unit (OBU)* que permite a comunicação entre os veículos em movimento ou com a infraestrutura através de *Road Side Units (RSUs)* que estão fixas nos meios urbanos, em pontos estratégicos, por exemplo, nos semáforos ou cruzamentos. A comunicação ocorre através de interfaces *wireless*, existindo assim dois tipos de comunicação: *Vehicle to Vehicle (V2V)* e *Vehicle to Infrastructure (V2I)*. Estes dois tipos de comunicação estão presentes na arquitetura de uma VANET representada na Figura 2.1:

- **Comunicação entre Veículos (V2V):** existe comunicação apenas entre os veículos, não havendo com a infraestrutura, isto é, os nós vão disseminando a informação em *multi hop* (através de veículos que estão pelo meio) até chegarem ao destino (Figura 2.1 (a)). Isto acontece quando o destino não está ao alcance da fonte. Este tipo de comunicação pode ser usada em aplicações de segurança e disseminação.
- **Comunicação entre Veículos e Infraestrutura (V2I):** neste caso, a comunicação faz-se entre OBUs e RSUs, isto é, os veículos comunicam com a infraestrutura (Figura 2.1 (b)). Este tipo de comunicação permite alcançar outras redes e pode ser usada para aplicações de recolha de informação.
- **Arquitetura Híbrida:** engloba os dois tipos de comunicações (V2I e V2V) (Figura 2.1 (a) e (b)). Cada veículo pode comunicar com a infraestrutura diretamente (um salto) ou através de vários saltos. Permite ligações à *Internet* a longas distâncias.

- **Comunicação entre as Infraestruturas (I2I):** este tipo de comunicação é utilizado pelas RSUs, nomeadamente em aplicações de gestão de tráfego (Figura 2.1 (c)).

Este trabalho considera a comunicação entre veículos (V2V), uma vez que o objetivo é utilizar apenas este tipo de comunicação para avisar a ocorrência de um acidente, num cenário de auto-estrada.

As VANETs apresentam características particulares tais como: topologia dinâmica, os veículos movem-se com velocidades na ordem dos 50 Km/h nas cidades e mais de 100 Km/h nas auto-estradas. Além disso, podem mudar a sua direção, fazendo com que os veículos permaneçam na rede por curtos períodos de tempo. Isto leva a que as ligações entre os veículos facilmente se percam, enquanto estão a transmitir informação entre eles e à dependência do conhecimento da localização geográfica dos veículos para os quais a informação vai ser enviada. Apesar da mobilidade e previsão reduzida, os veículos seguem padrões (estradas, velocidades limite, semáforos, entre outros); se estes forem conhecidos é mais viável prever a posição futura dos veículos.

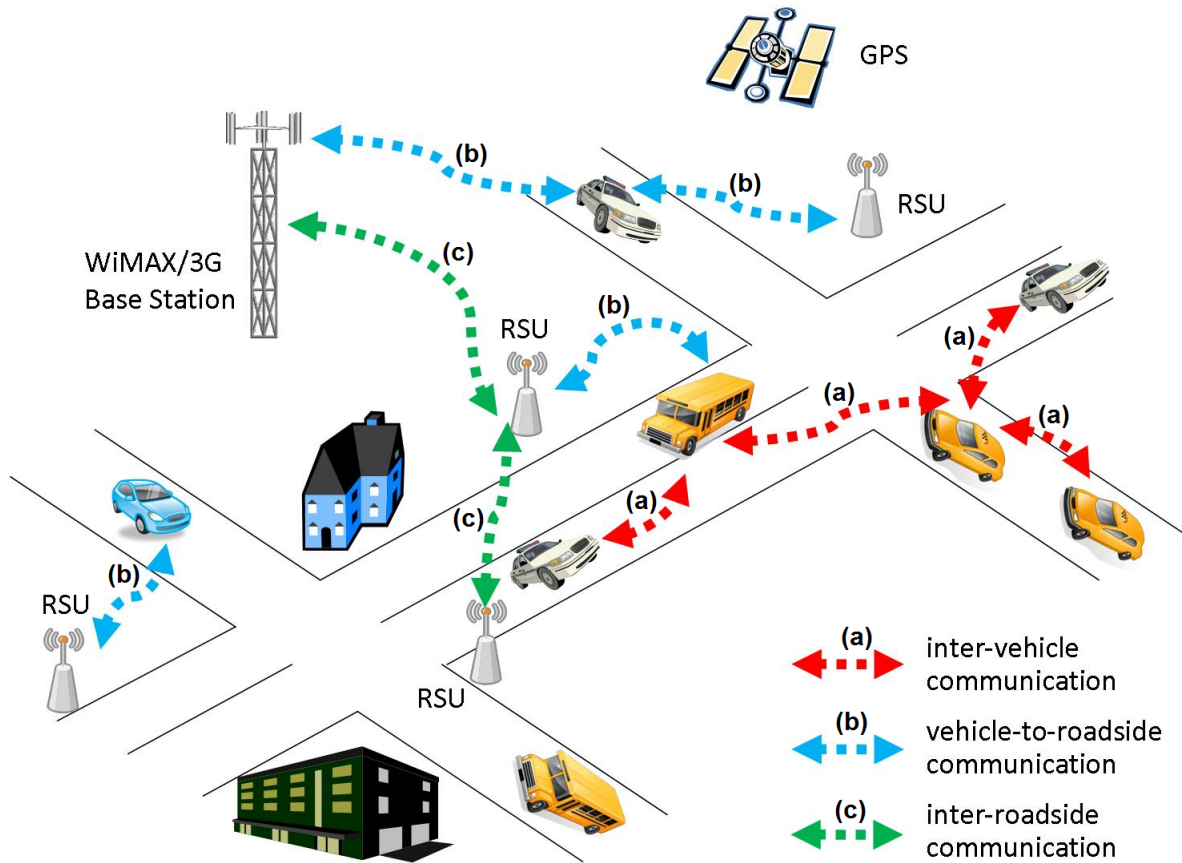


Figura 2.1: Arquitetura de uma VANET [4]

2.2.1 Disseminação de Dados

Os protocolos de encaminhamento, implementados na camada de rede são responsáveis por desenvolver estratégias que permitam comunicações de confiança sem que ocorram disrupções.

As VANETs permitem vários tipos de estratégias para a disseminação de dados.

A disseminação de qualquer tipo de dados pode ser feita recorrendo a várias estratégias: *unicast*, *multicast* e *broadcast*.

- **Unicast:** A comunicação é feita, tendo em conta que um nó de origem envia informação para um único nó destino, através de *multi hop*.
- **Multicast/Geocast:** A informação é enviada de um nó origem para um grupo específico de nós. *Geocast* é um tipo de comunicação *multicast* cujo destino é um conjunto de nós bem definido, por exemplo, uma área geográfica que é, normalmente, definida em relação à localização do nó origem.
- **Broadcast:** A informação é enviada de um nó origem para todos os nós ao mesmo tempo, que estiverem ao alcance do nó origem, como é o caso das mensagens de emergência. Por sua vez, os nós que receberam a mensagem vão reenviar para todos os seus vizinhos, até a mensagem chegar aos nós destino.

Para além das estratégias referidas anteriormente, a disseminação pode ser classificada quanto ao número de saltos necessários para que a informação chegue desde a fonte até ao seu destino:

- **Um Salto (*Single hop*):** A informação é encaminhada através de nós diretamente ligados.
- **Vários Saltos (*Multi hop*):** A informação é encaminhada recorrendo a nós intermédios até chegar ao seu destino, isto é, o destino não está ligado diretamente nem ao alcance da fonte.

2.2.2 Pilha de Protocolos para VANETs

Uma vez que as VANETs permitem comunicações V2V e V2I, a pilha de protocolos tem de lidar com as características destes dois tipos de comunicações. Ora, como as VANETs não têm uma configuração prévia, existem vários desafios no que diz respeito ao projeto dos protocolos.

Em 1999 a *Federal Communication Commission (FCC)* atribuiu aos sete canais (172, 174, 176, 178, 180, 182 e 184), uma largura de banda de 75 MHz, correspondente à gama de frequências: 5.850-5.925 GHz. No entanto, a *European Telecommunications Standards Institute (ETSI)* atribuiu uma largura de banda de 70 MHz, que corresponde à gama de frequências: 5.855–5.925 GHz. Esta gama destina-se exclusivamente a comunicações V2V e V2I. Como se pode ver pela Figura 2.2, existem sete canais, cada um com uma largura de banda de 10 MHz, que podem ser divididos em dois tipos: 6 Canais de Serviço (CSs) e 1 Canal de Controlo (CC). De acordo com a ETSI o espetro das frequências está dividido consoante o tipo de aplicações a que se destina. Ainda na Figura 2.2 é possível ver essa divisão: entre 5.855 MHz e 5.875 MHz destina-se a aplicações *Intelligent Transportation Systems (ITS)* não destinadas a segurança, entre 5.875 MHz e 5.905 MHz destina-se a aplicações de segurança e eficiência no tráfego, e entre 5.905 MHz to 5.925 MHz destina-se a futuras aplicações em *ITS*. Sempre que o CC está ativo, todos os nós param as suas comunicações, enquanto o canal estiver a receber e transmitir mensagens de segurança.

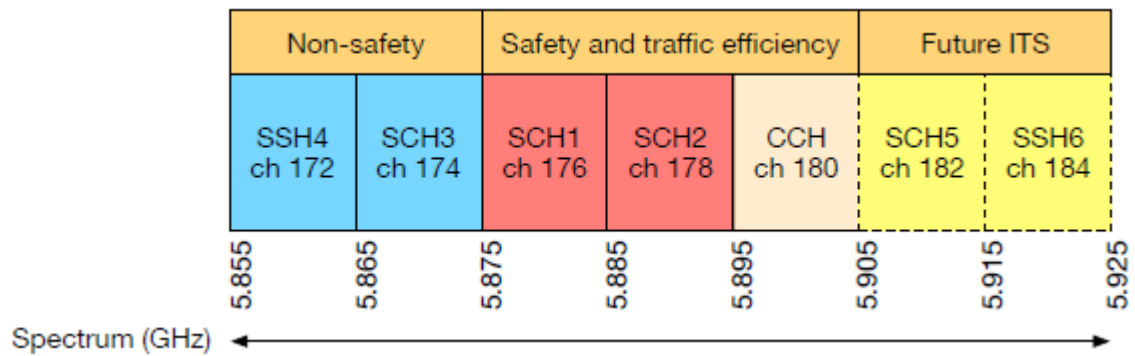


Figura 2.2: Canais e Espectro da Largura de Banda do DSRC de acordo com a norma Europeia IEEE 802.11p [5]

2.2.2.1 IEEE 802.11p

O IEEE desenvolveu o protocolo WAVE (IEEE 802.11p) com o objetivo de ser usado e implementado em Redes Veiculares. De acordo com Jiang and Delgrossi [6], o WAVE é capaz de lidar com as características especiais e os desafios que uma Rede Veicular apresenta, como a mudança muito rápida da rede, uma vez que os nós estão em constante movimento e cujas ligações são de curta duração, e permite a troca de mensagens sem recorrer a uma *Basic Service Set (BSS)* (ao contrário do IEEE 802.11 tradicional). Foram feitas algumas alterações na camada física (*Physical Layer*) relativamente ao IEEE 802.11a tais como: a mudança da largura de banda dos canais de 20 MHz para 10 MHz, a frequência de 5 GHz para 5.9 GHz e a máscara de transmissão também foi melhorada em relação ao IEEE 802.11a, tendo sido criadas 4 máscaras destinadas a 4 classes de operação. Este protocolo define a técnica de sinalização e função das interfaces, controladas pela camada *Medium Access Control (MAC)*.

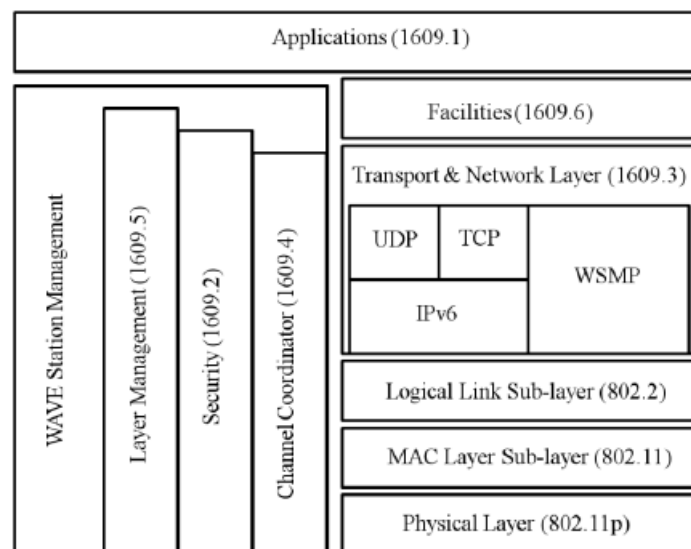


Figura 2.3: Protocolo WAVE [7]

2.2.2.2 IEEE 1609.x

A família IEEE 1609.x é constituída por um conjunto de 4 padrões que especificam vários serviços e aplicações fornecidos pela camada MAC. Segundo Gukholl and Cherkaoui [8], cada um dos 4 padrões é responsável por uma área específica:

- IEEE 1609.1 [9] é o *core* da arquitetura do WAVE e corresponde às camadas superiores do modelo de *Open Systems Interconnection (OSI)* [10], permitindo especificar os tipos de dispositivos que as OBUs suportam. São definidos aqui os fluxos dos recursos e dados, os formatos das mensagens de comando e de armazenamento dos dados.
- IEEE 1609.2 [11] é o responsável pelos serviços de segurança. Define o formato, o processamento e as circunstâncias em que as mensagens de segurança são trocadas.
- IEEE 1609.3 [12] é o responsável pelos serviços de Rede, correspondentes às camadas de rede e transporte do modelo de OSI [10], tratando do endereçamento e do encaminhamento necessário para a troca de dados WAVE seguras. Além disso, também descreve o protocolo *WAVE Short Message Protocol (WSMP)*, que é uma alternativa ao Internet Protocol (IP) e define um conjunto de funções de gestão implementadas no *WAVE Management Entity (WME)*.
- IEEE 1609.4 [13] é o responsável pela gestão dos canais e permite o uso em simultâneo dos dois tipos de canais, CS e CC e corresponde à segunda camada do modelo de OSI [10].

2.2.3 Aplicações e Serviços

As aplicações de uma VANET podem classificar-se em várias categorias: segurança, eficiência, conforto e entretenimento [14]. Neste trabalho, o foco são as aplicações relacionadas com situações de emergência, nomeadamente o aviso da ocorrência de acidente numa auto-estrada.

- **Aplicações de Eficiência:** Estas aplicações têm como objetivo melhorar a mobilidade dos veículos que estão na via pública. Utilizam dois tipos de comunicação: V2V e V2I. As aplicações de eficiência podem ser divididas em aplicações que permitem reduzir e evitar os engarrafamentos e controlar o trânsito nos cruzamentos e interseções.
- **Aplicações de Conforto:** Nestas aplicações os condutores podem receber informações que ajudem e tornem a viagem mais confortável. Estas informações devem estar disponíveis na altura em que o condutor precisa, de forma a permitir tomar decisões atempadamente sem prejudicar a viagem. Este tipo de aplicações englobam informações relativas ao tempo, localização de postos de abastecimento ou de restaurantes, informações turísticas, entre outras.
- **Aplicações de Entretenimento:** Estas aplicações destinam-se a fornecer informações relacionadas com entretenimento aos condutores e passageiros. Essas informações deviam ser adaptadas ao contexto dos utilizadores, mas como existe uma constante mobilidade dos veículos e das pessoas, torna-se complicado manter essas informações atualizadas. As comunicações usadas são normalmente V2V e V2I. Alguns exemplos de aplicações de entretenimento são: o acesso à *Internet*, assistir a vídeos ou descarregar uma música, partilha de ficheiros ou até mesmo jogar um jogo *online*.

- **Aplicações de Segurança:** O principal objetivo deste tipo de aplicações em VANETs é diminuir ou evitar o número de acidentes rodoviários. Para reduzir o atraso a que estas aplicações estão sujeitas, são usadas comunicações V2V. Este tipo de aplicações envolvem situações de emergência, onde o tempo de reação, geralmente, é muito baixo. Estas situações requerem a comunicação entre veículos e dos mesmos com a infraestrutura (caso não existam veículos ao alcance), sendo que, neste trabalho vai ser utilizada exclusivamente, a comunicação V2V. Várias situações podem ser classificadas neste tipo de aplicações tais como: evitar um acidente, avisar a ocorrência do mesmo, perigo de ultrapassagem, excesso de velocidade na aproximação de uma curva apertada ou com pouca visibilidade e uma travagem brusca.

2.2.4 Desafios

Tendo em conta as características das VANETs referidas anteriormente, vão ser apresentados, de seguida, alguns desafios nestas redes no futuro, tais como:

- **Redes Veiculares altamente heterogéneas:** A mudança rápida da topologia da rede faz com que o endereçamento dos nós, a qualidade de serviço, a segurança e os protocolos de encaminhamento sejam complexos. Portanto, é expectável que a próxima geração de ITS tenha uma visão mais global sobre as soluções de rede.
- **Gestão e armazenamento de informação:** Num futuro próximo existirão redes com milhões de veículos, que, por sua vez, irão gerar grandes quantidades de informação que deverá ser guardada de alguma forma e de forma distribuída através das VANETs.
- **Sistemas de Localização:** Em aplicações críticas de segurança é fundamental conhecer a localização dos veículos. Uma solução comum seria a integração de um sistema de GPS em cada carro. No entanto, estes sistemas têm alguns problemas como: a fraca intensidade do sinal em túneis, que prejudica a receção das coordenadas, ou a imprecisão da posição dos veículos numa ponte (em cima ou por baixo da mesma). Existem diversas técnicas de localização denominadas: *Map Matching*, *Dead Reckoning*, *Cellular Localization*, *Image/Video Processing*, *Localization Services*, *Differential GPS technique* e *Relative Distributed Ad Hoc Localization* [15]. Todas estas técnicas têm vantagens e desvantagens, mas nenhuma sozinha é capaz de satisfazer todas as condições exigidas pelas VANETs. Por esse motivo será necessário uma combinação de diferentes técnicas que permitam ter um sistema de localização confiável. Dado que a mudança da topologia é muito frequente em VANETs, isto faz com que a informação disseminada esteja desatualizada relativamente às posições dos veículos. Portanto, o estudo de modelos que façam uma previsão das posições futuras ao longo do tempo será uma boa alternativa, no futuro, aos sistemas de localização.
- **Comunicações tolerantes a interrupções:** Os atrasos longos e interrupções, são problemas bem conhecidos das VANETs. De forma a tentar resolver estes problemas é muito frequente a utilização do mecanismo *Store-Carry-and-Forward* que permite que a informação seja entregue ao destino. No futuro deverão ser estudadas novas abordagens de comunicação de dados para VANETs.
- **Endereçamento Geográfico:** É necessário conhecer a posição ou área geográfica dos veículos, de forma a ser possível realizar comunicações e enviar dados. Portanto, dado

o padrão do movimento dos veículos e sabendo o comportamento dos condutores, a previsão da posição futura do veículo torna-se um desafio que é importante no caso das VANETs.

- **Rastreo do alvo (veículos):** Como já foi referido anteriormente, é fundamental conhecer a posição futura do veículo para aplicações que necessitem de fazer o rastreo dos veículos na rede e também para os protocolos de encaminhamento.
- **Uniformização dos protocolos:** Existem diversos tipos de veículos numa VANET como, carros, autocarros, camiões, táxis, entre outros. De forma a ser possível existir comunicação entre eles, é necessário uniformizar o tipo de protocolo usado para as comunicações.
- **Densidade Variável da Rede:** Uma das características de uma VANET é a variação da densidade da rede. Os veículos têm de adaptar o seu comportamento à densidade do cenário em que se encontram, de forma a não prejudicar o envio e receção de dados.
- **Fragmentação da Rede:** A fragmentação da rede é outra característica bem conhecida das VANETs. Este tipo de problema pode levar a que alguns nós fiquem isolados ou inacessíveis e ocorrem em cenários de baixa densidade, como é o caso das áreas rurais. No futuro será necessário criar protocolos que não dependam das informações da topologia da rede.

Nesta dissertação pretende-se resolver alguns destes desafios, como por exemplo, a comunicação de mensagens de emergência com densidade variável das redes veiculares, o armazenamento e gestão da informação e desafios relacionados com a localização dos veículos, recorrendo a três estratégias de disseminação de mensagens de emergência numa VANET.

2.3 Protocolos de Encaminhamento aplicados a VANETs

Existem diversas categorias de protocolos de encaminhamento aplicados a VANETs e que são aplicáveis em cenários de emergência, tais como o aviso ou prevenção de um acidente, congestionamento do tráfego, entre outros. A Figura 2.4 apresenta as seis categorias em que são classificados os protocolos de *Routing*. Os protocolos baseados na topologia [16] escolhem a rota desde a origem até ao destino através das informações guardadas previamente pelos veículos. A manutenção da rota é obrigatória antes de enviar os pacotes de dados. Os protocolos baseados na posição [16] selecionam os nós que vão enviar a mensagem através da posição geográfica dos veículos. Neste caso não é necessário conhecer a rota para enviar os pacotes. Os protocolos baseados em *clusters* [17] agrupam os veículos com características semelhantes (direção e velocidade, por exemplo) e selecionam um *Cluster Head (CH)* que é um nó (pertencente ao *cluster*) responsável por enviar os pacotes por comunicar com os restantes *clusters*. Os mais comuns são os protocolos de *broadcast* [18], onde todos os nós enviam os pacotes para todos os vizinhos, com o objetivo de alcançar todos os veículos. Por fim, os protocolos baseados na infraestrutura [17] utilizam as RSUs situadas nos cruzamentos ou ao longo da estrada para encaminhar os pacotes para os veículos que estiverem dentro da sua área de alcance.

Nesta secção vão ser apresentados três exemplos de protocolos pertencentes à categoria *Geocast* (dois deles utilizam *beacons*, *Inter-Vehicles Geocast (IVG)* e *Distributed Robust Geocast (DRG)* e outro que não depende de *beacons*, *RObust VEhicular Routing (ROVER)*). Vai ser dada ênfase a esta categoria, uma vez que estes protocolos partem do princípio que a informação é enviada por um único veículo origem, para todos os veículos que se encontram numa área específica denominada Zona de Relevância. Essa estratégia é semelhante à adotada na solução proposta nesta dissertação.

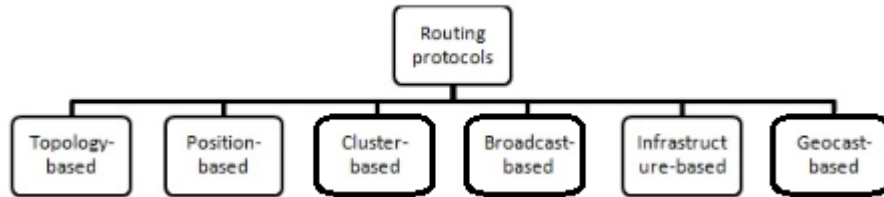


Figura 2.4: Classificação dos protocolos de encaminhamento em VANETs [19]

2.3.1 Protocolos *Broadcast*

Este tipo de protocolo usa a técnica de *flooding* (“inundam” a rede), isto é, geralmente todos os veículos enviam pacotes para todos os seus vizinhos, tendo como objetivo alcançar todos os veículos pertencentes à rede. São usadas algumas técnicas de seleção de nós que reenviam os pacotes, de forma a reduzir a quantidade de pacotes redundantes que circulam na rede. Esta questão do excesso de pacotes redundantes pode constituir um problema, tendo em conta cenários de alta densidade, isto é, uma elevada quantidade de veículos por Km^2 .

2.3.2 Protocolos *Geocast*

Estes protocolos são caracterizados pelo encaminhamento de informação utilizando a localização geográfica dos veículos, isto é, as mensagens são enviadas para todos os veículos contidos numa região bem definida, ZoR. A escolha destes exemplos de protocolos prende-se com o facto mencionado anteriormente, utilização e não utilização de *beacons*, e por serem citados na literatura que diz respeito a este tipo de protocolos.

- **ROVER:** O objetivo deste protocolo assenta numa base confiável de um protocolo de transporte que permita aplicações numa VANET que necessitem do acesso à *Internet*. O ROVER [20] usa um processo de descoberta de rota dentro de uma dada ZoR semelhante ao *Ad hoc On-Demand Distance Vector (AODV)* [21]. Neste protocolo, apenas os pacotes de controlo são enviados para todos os vizinhos, uma vez que os pacotes de dados são enviados em *unicast*, isto é, para os nós que estão a um salto. Ora, como os pacotes de dados são enviados para os vizinhos que estão a um salto e como todos os veículos possuem GPS, é necessário enviar *beacons* periódicos para que o nó fonte conheça os seus vizinhos.

A Figura 2.5 mostra os pressupostos que o protocolo ROVER assume:

- cada veículo possui um Identificador (ID) único;

- todos os veículos estão equipados com GPS;
- todos os veículos têm acesso a um mapa;
- a ZoR é definida como um retângulo;
- a *Zone of Forwarding (ZoF)* é um retângulo que engloba a ZoR e o nó fonte que inicia o envio dos pacotes.

A mensagem é definida como um tripleto $[A, M, Z]$, em que **A** representa uma aplicação, **M** a mensagem que vai ser enviada através da aplicação, e **Z** a ZoR onde a mensagem vai ser disseminada.

O objetivo do processo de descoberta de rotas é construir uma árvore desde o nó fonte e que englobe todos os veículos contidos na ZoR. A Figura 2.5 mostra um exemplo desse processo, que é iniciado quando o nó fonte envia a mensagem, *Zone Route REQuest (ZRREQ)* para todos os vizinhos na ZoF, contendo informações relevantes sobre si e sobre a ZoR atual. Os nós que estiverem dentro da ZoR guardam o pacote e respondem ao nó que estiver a um salto que enviou a ZRREQ com uma mensagem *Zone Route REply (ZRREP)* que contém o ID do nó. Os nós que estiverem dentro da ZoF, mas fora da ZoR, não respondem com uma mensagem ZRREP. Todos os nós guardam as informações necessárias para construir uma árvore *multicast* até ao nó fonte. Neste caso, a distância ao nó origem é calculada da seguinte forma: $\alpha * R$, onde **R** é o alcance máximo de transmissão e α varia entre 0 e 1. Esta informação é importante, pois permite construir uma árvore mais robusta. Desta forma, cada nó conhece o próximo salto e envia a mensagem para o próximo nó que está a um salto (*unicast*), e assim a informação vai seguindo a árvore construída (como se vê na imagem da direita da Figura 2.5). À medida que os nós se vão deslocando, todas as informações acerca da ZoR e ZoF vão sendo alteradas e guardadas numa tabela, de forma a ser possível ir modificando a rota que constitui a árvore *multicast*.

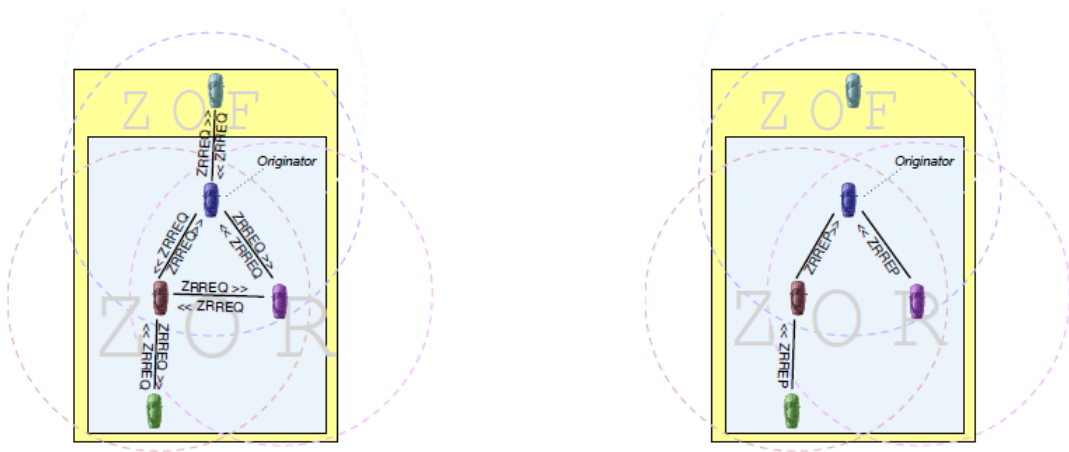


Figura 2.5: Processo de descoberta de rotas usando mensagens ZRREQ e ZRREP [20]

- **DRG:** O objetivo deste protocolo baseia-se num encaminhamento rápido e de confiança de mensagens que minimize a carga da rede. O DRG [20] não utiliza o conhecimento

do estado atual da rede, no entanto é capaz de se adaptar às mudanças constantes de topologia da mesma. O seu algoritmo é baseado na distância entre os veículos, numa estrada de uma ou duas dimensões. Além disso, possui mecanismos que conseguem ultrapassar a situação temporária de inexistência de nós para reenviar a mensagem e prevenir ciclos, isto é, que a mensagem volte a ser enviada para o nó de quem recebeu.

O algoritmo de encaminhamento do DRG escolhe os nós que estão, sucessivamente, mais próximos do destino para reenviar a mensagem, isto é, quando o primeiro nó envia a mensagem, o nó que estiver no limite da sua área de transmissão é o escolhido para disseminar a informação em *broadcast* (Figura 2.6). Assim que um nó recebe uma mensagem, calcula um tempo de espera antes de a retransmitir, que é inversamente proporcional à sua distância em relação ao último nó que enviou a mensagem. Desta forma, percebe-se que o nó que estiver a uma maior distância será o escolhido para retransmitir, pois quanto maior for a distância, menor será o tempo de espera. Portanto esse nó irá retransmitir e os restantes cancelarão as suas retransmissões.

A fórmula usada para o cálculo desse tempo é:

$$BO_d(R_{tx}, d) = MaxBO_d * S_d\left(\frac{R_{tx} - d}{R_{tx}}\right) \quad (2.1)$$

onde BO_d representa o tempo de espera que depende da distância ao último nó que transmitiu, $MaxBO_d$ é o tempo máximo permitido de espera de cada nó, S_d é o fator de sensibilidade da distância para ajustar o tempo de espera, R_{tx} é o alcance da transmissão e d é a distância ao último nó transmissor.

Este tipo de seleção faz com que não seja necessário a troca periódica de *beacons*, pois o processo de troca periódica de *beacons* leva a uma elevada quantidade de pacotes redundantes na rede.

Quando existe fragmentação da rede, isto é, quando não existe um nó dentro do alcance de transmissão, um dos mecanismos usados é a retransmissão periódica feita pelo nó que se encontra no limite da área de transmissão até encontrar um novo nó que retransmita a mensagem.

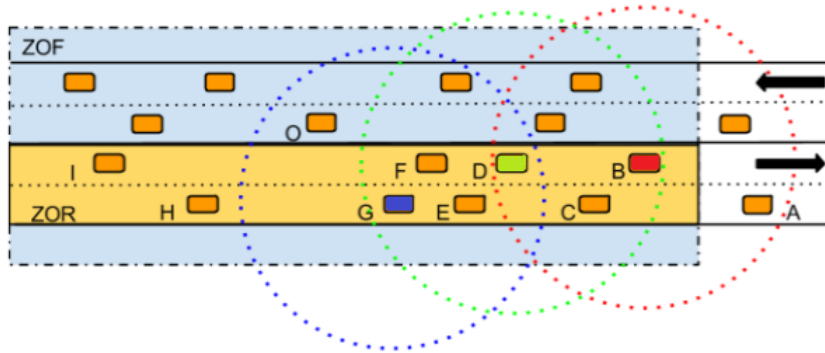


Figura 2.6: Cenário de estrada usando o método baseado na distância [22]

Quando o cenário é de duas dimensões, ou seja, um cenário de cidade onde existem várias interseções, a escolha dos nós muda significativamente. Os nós selecionados serão aqueles

que estiverem melhor posicionados, de forma a cobrirem novas áreas de transmissão da ZoR. Os nós continuam a retransmitir a mensagem até receberem uma confirmação de outros nós que também estão a retransmitir, de forma a que a eficiência da entrega e propagação sejam elevadas. No entanto, se o nó atual receber uma mensagem repetida de outro nó que também está a transmitir, que cobre grande parte da sua área de transmissão, a probabilidade de outros nós que estão dentro dessa área de transmissão terem recebido essa mensagem é muito elevada.

- **IVG:** Este protocolo consiste em avisar os veículos que se encontrem num cenário de auto-estrada que ocorreu uma situação de perigo, como por exemplo, um acidente [23]. O conjunto de veículos que estiverem dentro da área de risco constitui um grupo, pelo que a informação é enviada para um grupo específico e bem caracterizado, *multicast*. Neste protocolo não existe o conceito de ZoR, mas sim áreas de risco, constituídas por veículos que podem, ou não, estar dentro dessas determinadas áreas, consoante a direção do movimento e a localização dos mesmos.

Quando um nó que está dentro da área de risco recebe uma mensagem de emergência, calcula um tempo de adiamento de retransmissão ($defertime(x)$) que depende da sua localização nessa área. Quando esse tempo acabar, ele decide se é relevante fazer a retransmissão ou não. Esse tempo é inversamente proporcional à distância entre o último nó que enviou a mensagem e os nós que estão dentro da sua área de transmissão, e é calculado através da seguinte equação:

$$defertime(x) = Maxdefertime * \left(\frac{R^\varepsilon - D_{zx}^\varepsilon}{R^\varepsilon} \right) \quad (2.2)$$

- R é o alcance de transmissão;
- D_{zx} é a distância entre os nós (z e x), por exemplo;
- $\varepsilon = 2$;

Portanto, o nó mais distante, x , tendo em conta a Figura 2.7 é o favorito para retransmitir, pois é o que terá o menor tempo de espera. Este tipo de seleção faz com que não seja necessário a troca periódica de *beacons*.

De seguida, serão apresentados alguns trabalhos relacionados baseados em *clusters*.

2.4 Delay Tolerant Networks (DTNs)

2.4.1 Definição

Segundo Ma *et al.* [24], uma DTN é definida como uma área da rede capaz de lidar com desafios tais como desconexões e interrupções da rede, sem uma conexão ponto-a-ponto. Foi feita para trabalhar com comunicações a longas distâncias (comunicações inter espaciais, por exemplo), cuja latência é um desafio pois neste tipo de ambientes podem ser horas ou até mesmo dias. Os protocolos de *Internet* são, normalmente baseados no protocolo *Transmission Control Protocol (TCP)/IP* [25] que não está preparado para trabalhar em ambientes com atrasos e interrupções na rede devido aos pressupostos fundamentais da *Internet*. Segundo Cerf *et al.* [26] os pressupostos da *Internet* baseiam-se em:

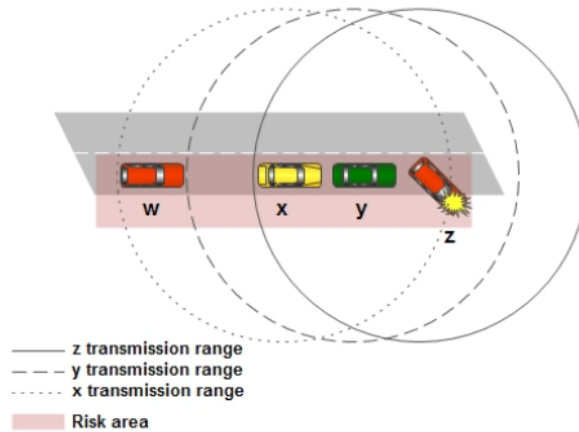


Figura 2.7: Processo de seleção dos nós retransmissores usando o protocolo IVG [20]

- Baixa probabilidade de perda de pacotes *end-to-end*;
- Todos os elementos da rede (*routers* e *endpoints*) suportam o protocolo TCP/IP;
- As aplicações não têm de se preocupar com o desempenho da comunicação;
- Durante a comunicação existe um caminho *end-to-end* entre a fonte e o destino;
- O protocolo TCP/IP fornece confiança nas comunicações, tanto na correção de erros como na gestão de congestionamento;

2.4.2 Arquitetura

De forma a ser possível garantir uma comunicação de confiança entre os nós quando a rede está sujeita a interrupções constantes, foi criada uma nova camada, designada *Bundle*, situada entre a camada de Transporte e a de Aplicação (Figura 2.8). Esta nova camada tem um armazenamento persistente que permite lidar com o problema das interrupções frequentes existentes na rede e um mecanismo de comunicação confiável que assenta num processo de confirmação *end-to-end*. Além disso, também tem outras funcionalidades como por exemplo, a gestão e segurança.

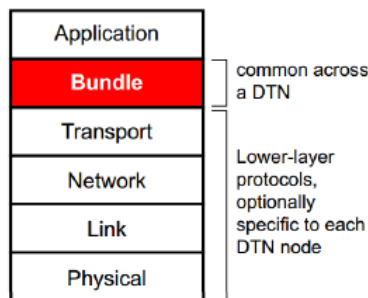


Figura 2.8: Protocolo DTN [27]

A DTN pode usar diversos tipos de protocolos de entrega que possuem diferentes características específicas e, portanto a DTN introduz um elemento novo denominado *Convergence Layer Adapter (CLA)*. O CLA é responsável por fornecer todas as funções necessárias para transportar os *bundles* até ao seu protocolo correspondente, como se verifica na Figura 2.9. Ainda na mesma figura é possível ver que existe um elemento central, *Bundle Forwarder*, que é o responsável por encaminhar *bundles* entre as aplicações, CLAs e fazer o armazenamento baseado em decisões de encaminhamento. Além disso, troca diretrizes usadas pelo processo de gestão, as aplicações e decisões de encaminhamento.

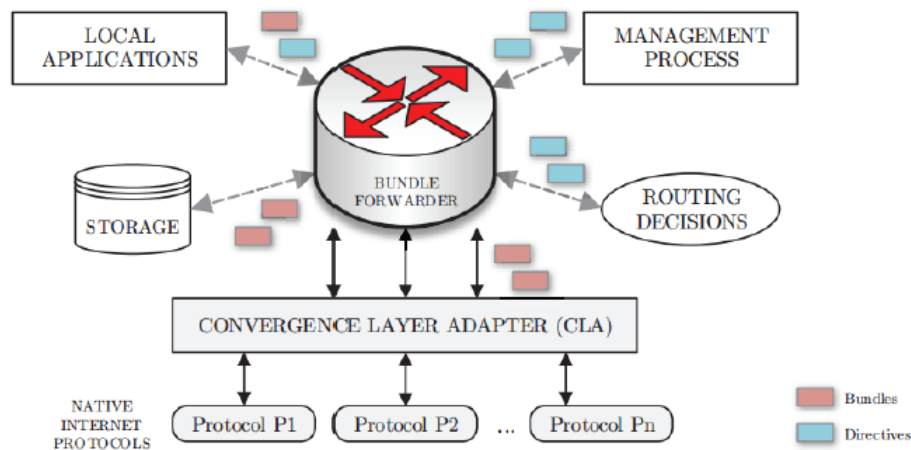


Figura 2.9: Arquitetura de Encaminhamento de um *Bundle* [28]

2.4.3 Mecanismo de SCF

Como já foi referido anteriormente, existem alguns problemas como a conectividade intermitente, taxas de erros elevadas e atrasos longos. De forma a combater esse tipo de problemas, as DTNs desenvolveram um mecanismo designado Store-Carry-and-Forward (SCF) que está representado na Figura 2.10. Uma vez que nem sempre existem ligações confiáveis e disponíveis, a DTN necessita de armazenar e transportar a informação e só a encaminha quando encontra uma ligação estável. Para conseguirem armazenar essa informação, os nós necessitam de estar equipados com dispositivos que permitam o armazenamento persistente da informação (por exemplo, disco rígido).

Segundo F. Warthman *et al.* [28] os nós precisam de ter armazenamento persistente devido às seguintes razões:

- A ligação ao próximo salto pode não estar disponível durante um longo período de tempo;
- Um par de nós pode ter uma taxa de transmissão discrepante, isto é, um nó pode enviar ou receber dados muito mais rápido do que o outro;
- Se após o envio da informação, esta for rejeitada pelo nó destino ou se ocorrer um erro, a informação deve ser retransmitida.

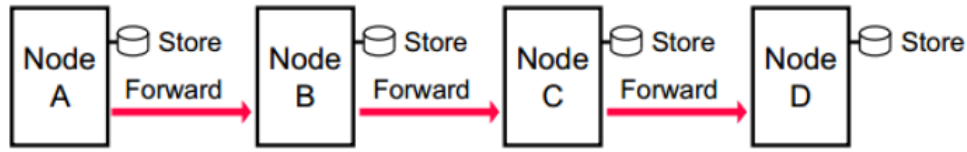


Figura 2.10: Mecanismo SCF [28]

2.5 Trabalhos Relacionados baseados em *clusters*

As mensagens de emergência são usadas para alertar um conjunto de situações de emergência numa VANET, como por exemplo, acidentes, más condições do piso da estrada, uma travagem brusca ou até mesmo uma ultrapassagem perigosa. Essas mensagens têm um conjunto de requisitos que as tornam diferentes das mensagens de conteúdos não urgentes. Esses requisitos são: alertar um conjunto de veículos que se encontram numa área específica ZoR, utilizar as coordenadas GPS de cada veículo para enviar a sua localização para todos os seus vizinhos e para definir os limites da ZoR, seleção de veículos, de acordo com a sua localização, que vão ter a função de disseminar a mensagem de emergência e a curta duração do processo de disseminação, uma vez que não fará sentido propagar a mensagem de emergência durante horas.

Tendo em conta alguns destes aspetos, Buchenscheit *et al.* [29] pensou, inicialmente, numa solução que prevenisse acidentes de veículos de emergência. Para tal, tira partido, sobretudo das comunicações V2V, mas também recorre à comunicação V2I (através dos semáforos). Os veículos recebem informações sobre a aproximação de um veículo de emergência e da rota, para, desta forma, os condutores reagirem atempadamente (por exemplo, mudarem de via) e evitar um acidente.

De acordo com Ramakrishnan *et al.* [30] "No futuro, apenas haverá comunicações entre veículos que irão alertar os condutores dos acidentes." Nesta dissertação, a disseminação de mensagens de emergência será feita, tendo em conta apenas as comunicações V2V.

Existem várias técnicas para disseminar mensagens de emergência, sendo que a técnica de *broadcast* [31, 32, 33] é a mais comum e menos eficiente de todas, pois todos os veículos enviam mensagens, congestionando a rede com pacotes redundantes. Nesta dissertação foi implementada uma estratégia de *broadcast*, semelhante à desenvolvida em [33], onde se assume que todos os veículos contidos na ZoR enviam um pacote para todos os seus vizinhos. Esta estratégia será usada como base para mostrar a eficiência das outras estratégias ao disseminar mensagem de emergência enquanto minimizam a troca de mensagens e consequentemente diminuem a sobrecarga na VANET.

Existem, no entanto, outras técnicas mais eficientes que minimizam a quantidade de pacotes redundantes na rede e número de saltos necessários para a mensagem chegar da fonte ao destino.

Os trabalhos em [30, 34, 35, 36, 37, 38, 39] propõem arquiteturas baseadas em agrupamentos de veículos, denominados *clusters* (Figura 2.11), aplicada a VANETs. Cada *cluster* contém: um único nó disseminador, designado Cluster Head (CH), um veículo que irá suceder o CH, designado de *Cluster Forwarder (CF)*, que é o nó mais afastado do CH e responsável por retransmitir os pacotes do CH, que se encontram dentro do mesmo *cluster* e fora do alcance do CH, sendo que os outros veículos receberão o pacote. Após saber a quantidade de veículos que constituem o *cluster*, o tamanho do mesmo depende da velocidade média dos

veículos. Esta velocidade vai ser comparada com um valor de velocidade limite predefinido e, se for menor, será formado um *cluster* de pequenas dimensões; caso contrário, será formado um *cluster* de maiores dimensões. Esta classificação das dimensões do *cluster* significa que é esperado que existam mais veículos cuja velocidade seja superior ao valor limite definido, e uma menor quantidade de veículos cuja velocidade seja inferior ao limite predefinido.

Em [30], a seleção do próximo CF é feita da seguinte forma: o CH contém duas listas ordenadas, *Cluster Black Lists (CBL)* e *Cluster Forwarder Lists (CFL)*, que vão sendo atualizadas consoante as mudanças na topologia. A primeira diz respeito aos veículos que fazem parte do *cluster*, cuja sua posição é inferior à do CH e está ordenada por ordem decrescente. A segunda contém os veículos pertencentes ao *cluster*, cuja sua posição é superior à do CH e está ordenada por ordem crescente. O CF é o nó mais afastado do CH pertencente à CBL.

As comunicações dentro de cada *cluster* ocorrem de forma a evitar colisões na rede, pois o CF envia as mensagens de emergência através do Canal de Controlo (CC), ao contrário das mensagens não urgentes que são transmitidas no Canal de Serviço (CS). O CH recebe as mensagens de emergência e envia-as a todos os nós do *cluster* através do CS. Foram introduzidas duas camadas MAC para transmitir pacotes de controlo, para aumentar a confiança no processo de transmissão das mensagens de emergência. Essas camadas designam-se *Request To Broadcast (RTB)* e *Clear To Broadcast (CTB)*.

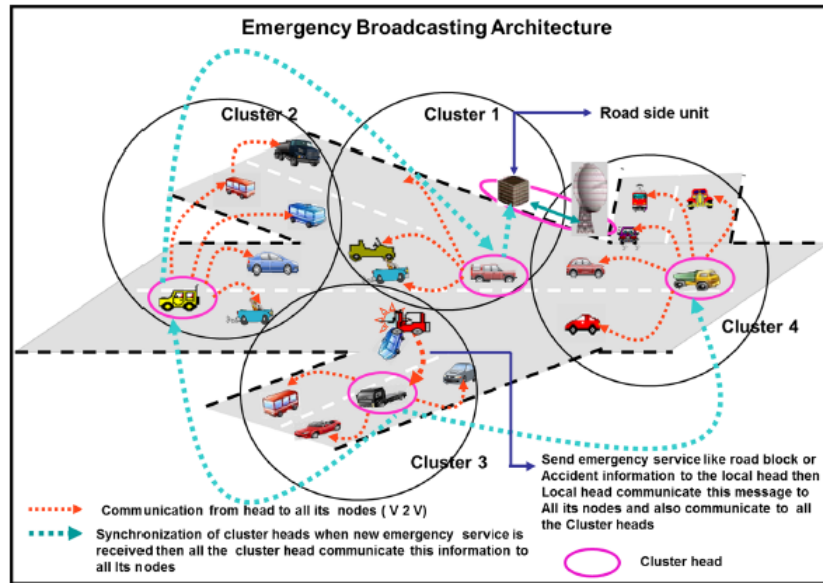


Figura 2.11: Arquitetura de *broadcast* de mensagem de emergência baseada em *clusters* para VANETs [30]

O modelo de auto-estrada baseado em *clusters* presente na Figura 2.12 proposto por Ramakrishnan *et al.* [40] pode ser aplicado a vários protocolos de encaminhamento considerando cenários de auto-estrada. O conceito de *cluster* permite melhorar a taxa de entrega dos pacotes de emergência, diminuir a redundância dos mesmos na rede e o atraso na propagação da mensagem desde a fonte até ao destino.

Portanto, a abordagem baseada no uso de vários *clusters* favorece a propagação em *multi hop* (Secção 2.2) entre veículos. Observando a Figura 2.12 vê-se, claramente, que os *clusters* facilitam a disseminação do pacote desde a fonte até ao destino, uma vez que a fonte envia

o primeiro pacote para o CH e este envia sucessivamente para os outros CHs até chegar ao destino (A-B-C-D). Ainda na Figura 2.12, se não forem considerados *clusters*, poderá acontecer perda de pacotes e atrasos na receção dos mesmos, devido à distribuição não uniforme dos veículos ao longo da auto-estrada. Isto significa que os pacotes seriam enviados para os sucessivos veículos da frente ou para os mais próximos em direção ao destino, aumentando o número de saltos e podendo levar à perda do pacote.

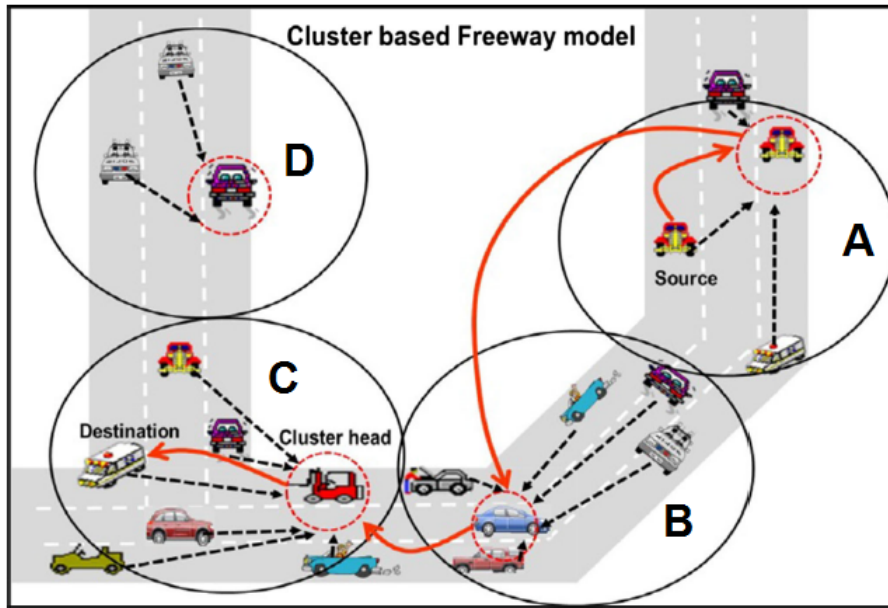


Figura 2.12: Arquitetura do modelo de auto-estrada baseada em *clusters* para VANETs [40]

Gokulakrishnan *et al.* [41] conceberam um modelo designado *Road Accident Prevention (RAP)* que serve para prevenir acidentes rodoviários em auto-estradas.

O processo de disseminação da mensagem de emergência começa numa RSU, que envia uma mensagem precoce aos veículos que estão na estrada, de forma a evitar potenciais acidentes. Este esquema, RAP é baseado num sistema de predição usando o protocolo *Emergency Situation Prediction Mechanism (ESPM)* [42]. Os veículos que se encontram na *Zona de Risco Elevado (ZRE)* recebem a mensagem instantaneamente, vinda da RSU.

Consoante seja a distância em relação ao acidente, os veículos são classificados em quatro áreas de risco: *Zona de Risco Elevado (ZRE)*, *Zona de Risco Médio (ZRM)*, *Zona de Risco Baixo (ZRB)* e *Zona Fora de Risco (ZFR)*.

Esta estratégia assenta numa estrutura *Vehicular Backbone Network (VBN)* [43, 44, 45, 46] que consiste na utilização de RSUs e OBUs, isto é, contempla dois tipos de comunicação, V2I e V2V. Esta estrutura está apresentada na Figura 2.13. A utilização de uma estrutura VBN tem algumas vantagens, tais como: redução do número de RSUs necessárias e, por conseguinte, cada RSU tem uma área mais extensa de cobertura (como se pode ver pelo retângulo a tracejado denominado *Extended Coverage Area of RSU B* da Figura 2.13), diminuindo a distância entre a fonte (RSU) e o destino, aumento da entrega das mensagens de emergência e diminuição do atraso *end-to-end*.

Existem dois tipos de nós que vão retransmitir a mensagem: as RSUs e os veículos que se encontram perto destas ou de uma RSU externa, isto é, o veículo está dentro do alcance de

outra RSU. Portanto, a seleção destes nós é feita consoante a sua velocidade e posição dos veículos em relação à fronteira com a área de cobertura das RSUs. Foram escolhidos estes critérios, pois quanto menor for a velocidade de um veículo, mais tempo permanecerá dentro da área de cobertura de qualquer RSU, e quanto mais perto um veículo estiver da fronteira entre as áreas de cobertura de duas RSUs, melhor será feita a disseminação da mensagem de emergência.

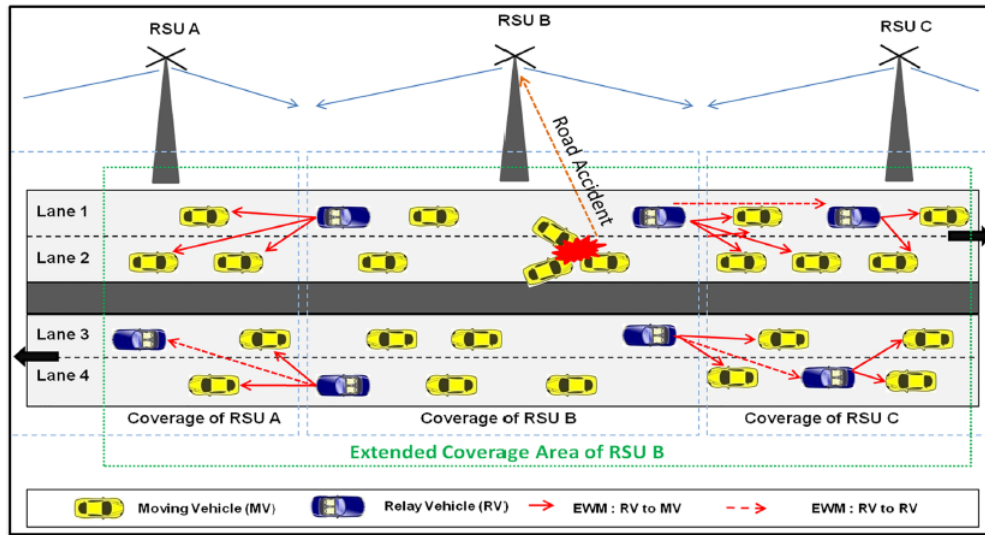


Figura 2.13: Arquitetura VBN aplicada a VANETs [41]

Nesta dissertação, foram implementadas duas estratégias de disseminação de mensagem de emergência (Alta e Baixa Densidade) cuja seleção dos nós que vão reenviar os pacotes é feita consoante a estratégia. No caso da estratégia de Alta Densidade, os *MaxRanks* (são os nós que vão retransmitir a informação) são os nós mais distantes que ainda estão dentro da área de transmissão do nó (*MaxRank*) que enviou previamente. No caso da estratégia de Baixa Densidade, os *MaxRanks* são selecionados consoante o número de vizinhos que tiverem atrás de si, sendo que o nó escolhido é o que tiver mais. Portanto, no caso da primeira estratégia, o conceito assemelha-se ao modelo de auto-estrada baseado em *clusters* proposto por Ramakrishnan *et al.*, mas a seleção dos nós que irão disseminar a mensagem obedece a critérios diferentes. Em relação à segunda estratégia, se se assumir que o tamanho do *cluster* está relacionado com a quantidade de vizinhos atrás que cada nó tem, poderá ser semelhante à estratégia que relaciona a velocidade dos veículos com o tamanho dos *clusters*. Portanto, ambas as estratégias se assemelham a protocolos baseados em *clusters*, mas com definições de *cluster* diferentes. Quanto à estratégia de *Broadcast*, é restringida uma área (ZoR) onde são enviados os pacotes e apenas os veículos que estão dentro da ZoR podem enviar pacotes de emergência.

2.6 Considerações do Capítulo

Neste capítulo foi descrito de uma forma sucinta o conceito e arquitetura de uma VANET, os tipos de disseminação existentes quanto aos saltos (um salto, vários saltos) e quanto à estratégia de comunicação (*unicast*, *multicast* e *broadcast*) e as aplicações de uma VANET.

Estas aplicações permitem minimizar algumas situações de emergência do cotidiano, como acidentes ou atropelamentos de peões, bem como informar os veículos quando algumas destas situações ou outras menos urgentes ocorrem. Também podem ser usadas para eventos considerados não urgentes, por exemplo, o entretenimento (informações turísticas, assistir a vídeos dentro do carro). Também foi apresentada a pilha de protocolos para VANETs, nomeadamente o protocolo WAVE que é o que vai ser usado nesta dissertação para as comunicações entre os veículos, isto é, disseminar as mensagens de emergência dentro da ZoR. De seguida, foram descritos alguns protocolos de encaminhamento aplicados a VANET baseados em *geocast* (IVG, ROVER e DRG) uma vez que, neste trabalho, a disseminação de conteúdos urgentes está diretamente ligada com as coordenadas GPS e *broadcast*. Por fim, foram apresentados alguns trabalhos relacionados com a disseminação de mensagens de emergência (baseados em *clusters*) e uma breve conclusão sobre as estratégias implementadas.

Capítulo 3

Estratégias para a Disseminação de Mensagens de Emergência

3.1 Descrição do Capítulo

No capítulo anterior foram abordados alguns assuntos relacionados com redes veiculares (VANETs), tais como: arquitetura, vários tipos de disseminação de dados, desafios, aplicações e Serviços. Além disso, foi apresentado e descrito o mecanismo de SCF existente nas DTNs, que permite lidar com a disrupção da rede presente nas VANETs. Foram ainda descritos alguns protocolos de encaminhamento, aplicados a VANETs e relacionados com esta dissertação: protocolos baseados em *Clusters*, posição geográfica (*Geocast*) e *Broadcast*.

Neste capítulo vai ser descrito e apresentado o problema proposto nesta dissertação, bem como as estratégias de disseminação de mensagens de emergência desenvolvidas para encaminhar informação urgente, numa dada área bem definida denominada ZoR. As três estratégias propostas têm conceitos semelhantes; no entanto, o método de seleção dos nós retransmissores (*MaxRanks*) varia em cada uma delas, sendo que a última não contém o método de seleção de nós retransmissores. Em cada estratégia serão explicadas as lógicas de envio e receção de pacotes de dados e de *advertisements*, e os diferentes métodos de seleção de nós retransmissores, utilizando diagramas de fluxo para melhor compreensão dos mesmos.

3.2 Problema

As VANETs apresentam características que se tornam em desafios para os processos de disseminação de mensagens de emergência. A mudança muito rápida da topologia da rede e o tempo reduzido de contacto dificultam a disseminação de qualquer tipo de mensagens. No entanto, as mensagens de emergência são mais críticas e necessitam de ser entregues num tempo curto e ter ligações estáveis, sem disrupções. Ora, por vezes existem disrupções nas comunicações que impedem a mensagem de chegar ao destino, sendo então fundamental o mecanismo SCF existente nas DTNs. Este mecanismo permite guardar a informação até existir uma ligação de confiança para enviar a mensagem até ao destino. Outra questão importante é a localização dos veículos, pois o envio das mensagens de conteúdo urgente depende das coordenadas geográficas de cada veículo, de forma a ser possível, por exemplo, delimitar uma zona onde estas vão ser disseminadas. Por outro lado, o problema da fragmentação da rede pode provocar a perda de pacotes, o que não é desejável no envio e receção de mensagens de

Tabela 3.1: Características dos vários protocolos [47]

Protocolo	Otimizações		
	Confiança	Redundância	Mensagens Prioritárias
Epidemic	-	Várias cópias	Não
DSCF	Não	Não	Não
FFRDV	<i>Hop ACK</i>	Não	Não
Infocast	-	Código	Não
C-DTN	Aberto	Aberto	Não
DvCast	Não	Não	Não
ROD	Não	Não	Não
UV-Cast	Não	Não	Não
SERVUS	Hop ACK	Várias cópias	Não

emergência. Em contrapartida, a densidade variável de veículos e a presença de edifícios ou obstáculos, leva a que as estratégias de encaminhamento tenham de se adaptar à densidade, de forma a tirar partido dos veículos e das respetivas localizações para que a informação chegue ao destino. Uma vez que também existem diferentes tipos de veículos (carros, camiões, autocarros, entre outros), é essencial uniformizar os protocolos de comunicação, possibilitando a utilização de todos os tipos de veículos e a comunicação entre os mesmos.

A Tabela 3.1 apresenta algumas características relativas a vários protocolos de disseminação. Em suma, o protocolo *Epidemic* é o que apresenta pior desempenho, pois tem um elevado número de pacotes redundantes, envia mensagens para todos os seus vizinhos de cada vez que efetua um contacto com um nó, originando problemas em cenários de alta densidade. Quanto aos restantes protocolos, a escolha dos nós retransmissores varia entre os protocolos, consoante a velocidade dos veículos ou a sua posição geográfica. A mudança entre os modos normal (*multi hop*) e SCF também está presente em todos os protocolos e varia conforme as estratégias de cada um. Em geral, o conceito de *clusters* para definir diferentes áreas onde os nós estão distribuídos na rede está presente em alguns destes protocolos (exceto *Epidemic*, *Infocast*, *Directional Store-Carry-and-Forward (DSCF)* e *Fastest Ferry Routing in DTN-enabled Vehicular Ad-hoc (FFRDV)*), pelo que nenhum deles dá prioridade às mensagens enviadas. Quanto ao impacto da confiança dos protocolos, isto é, a garantia que estes fornecem na entrega das mensagens, apenas é considerado por dois protocolos, FFRDV e SERVUS. Estes dois protocolos utilizam as mensagens *hop ACK* para garantir que a mensagem será entregue ao destino.

No caso das duas estratégias implementadas, que são as soluções propostas, existe alguma redundância que não pode ser ignorada, devido aos métodos de seleção dos nós retransmissores. Por exemplo, um dos métodos consiste na escolha do nó mais longínquo dentro do raio de transmissão WAVE do nó que está a transmitir, logo todos os nós que se encontram à frente dos nós transmissores vão receber mensagens repetidas. À semelhança do protocolo *UV-Cast*, as duas estratégias definem uma ZoR onde a mensagem de emergência vai ser disseminada e não contemplam priorização de mensagens, pois estas vão sendo sucessivamente enviadas à medida que são selecionados os nós.

Na sequência da disseminação de conteúdos não urgentes, surge a necessidade de expandir estes serviços já existentes para conteúdos urgentes, que têm como principais objetivos:

prevenção e informação de situações críticas de emergência, tais como um acidente, uma travagem brusca ou condições inadequadas do piso. O foco desta dissertação prende-se com a disseminação de mensagens de conteúdo urgente, numa situação de emergência, neste caso, um acidente, por todos os veículos que se encontram dentro da ZoR (atrás do acidente). Pretende-se fazê-lo através de um número mínimo de *hops*, isto é, reduzir o número de veículos que retransmitem essa mensagem, diminuindo o *overhead* na rede (e, consequentemente as retransmissões). A situação referida anteriormente vai ser estudada num ambiente de auto estrada, tendo em conta dois cenários distintos: alta e baixa densidade.

Por forma a ser possível implementar este novo conceito de disseminação de conteúdos urgentes, foi necessário alterar o módulo de encaminhamento existente no mOVERS, desenvolvida pelo grupo de investigação *NAP*, para projetos relacionados com Redes Veiculares. É neste módulo que são tomadas as decisões de envio e receção de pacotes, quer de dados, quer de *advertisements*, isto é, está dividido em duas lógicas: envio e receção dos mesmos. Na lógica de envio de pacotes de dados são definidos vários parâmetros, tais como: que tipo de nós e quais deles enviam a informação, através de que interface (WAVE, *Wireless Fidelity (Wi-Fi)*, por exemplo), as condições em que é realizado o envio de informação e para que nós é enviada essa dada informação. Na lógica de receção de pacotes de dados são definidos os seguintes parâmetros: qual a interface utilizada para receber a informação (WAVE, Wi-Fi, por exemplo), que tipo de nós e quais deles recebem e armazenam ou descartam uma determinada informação, os limites em termos de área onde os pacotes são disseminados e as condições em que é feito esse armazenamento ou descarte de informação de emergência útil. Na lógica de envio de pacotes de controlo são definidas as condições em que é realizado o envio dos mesmos, que tipo e quais os nós que fazem esse envio, qual o seu destino e quais as informações necessárias e úteis para a vizinhança, como coordenadas do acidente, o número de vizinhos que cada nó tem atrás e localização do próprio nó. Na lógica de receção de pacotes de controlo apenas são definidas as condições em que a receção é feita.

A Figura 3.1 representa o cenário que vai ser considerado e estudado ao longo deste trabalho. Será analisada uma fotografia no momento do acidente, isto é, os veículos encontrar-se-ão parados e será analisado apenas o instante do acidente. Através da Figura 3.1 é possível perceber que o pacote de emergência é enviado, inicialmente pelo veículo que sofreu o acidente e depois, sucessivamente pelos nós retransmissores (*MaxRanks*). A disseminação do pacote de emergência é feita apenas entre veículos, ou seja, V2V e o pacote é enviado sempre para todos os vizinhos.

Duas das três estratégias (Alta e Baixa Densidade) foram implementadas de forma a reduzir o número de pacotes repetidos que circulam na rede e a garantir que todas os veículos que se encontram na ZoR recebem o pacote gerado pelo veículo que teve o acidente. Desta forma, a eficácia na entrega dos pacotes de emergência é elevada e recorrendo a um número de saltos reduzido. A terceira estratégia (*Broadcast*) foi implementada sem ter em conta estes objetivos, tendo apenas uma diferença relativamente às outras duas que é a ausência de pacotes de controlo trocados entre os veículos. Esta estratégia vai ser utilizada para comparação com as duas estratégias propostas.

3.3 Disseminação de Conteúdos Urgentes - Solução Proposta

As estratégias de disseminação de mensagens de emergência consistem em reduzir o número de veículos que vão reenviar a mensagem, fazendo com que esta chegue igualmente a

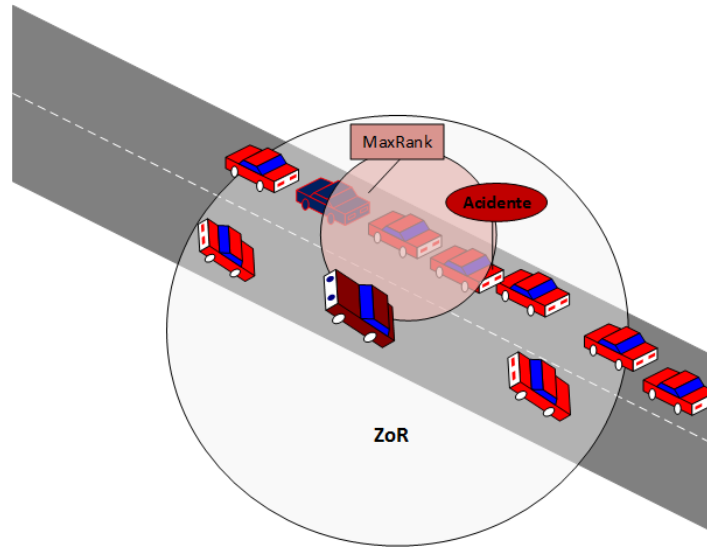


Figura 3.1: Cenário de estudo

todos os veículos que estão dentro da ZoR (circunferência centrada nas coordenadas GPS do veículo que teve o acidente com um determinado raio), preocupando-se apenas com os que se encontram atrás do acidente. Desta forma, o número de pacotes a circular na rede será reduzido, diminuindo o congestionamento da mesma, pois a escolha desses nós será feita através de uma seleção estratégica. A zona onde as mensagens de emergência vão ser disseminadas é uma área específica denominada ZoR e é determinada através das coordenadas do veículo que teve o acidente.

Desta forma, nesta dissertação são propostas duas estratégias que fazem uma seleção de veículos que farão a disseminação de mensagens de emergência por todos os seus vizinhos. Para fazer a escolha dos nós de disseminação, denominados *MaxRank*, as estratégias usam a posição dos nós e a sua direção de deslocamento, gerando um *rank* que varia consoante a estratégia pretendida.

Foram criadas duas estratégias com o objetivo de analisar dois tipos de cenários: alta e baixa densidade numa estrada.

3.3.1 Estratégia de Alta Densidade

A estratégia de disseminação de mensagens de emergência para alta densidade consiste em escolher sucessivamente o veículo vizinho mais afastado, mas cuja qualidade da ligação entre eles ainda seja válida, isto é, com o qual ainda consegue comunicar (com um Received Signal Strength Indicator (RSSI) acima de um valor predefinido) para retransmitir a mensagem até chegar a todos os veículos na ZoR. Inicialmente, o veículo que tem o acidente envia a mensagem para todos os seus vizinhos e calcula a sua distância relativamente a todos eles e escolhe aquele que está mais afastado de si para reenviar a mensagem para todos os seus vizinhos. O identificador do vizinho selecionado é colocado no cabeçalho do pacote de emergência para que esse vizinho saiba que é ele o próximo *MaxRank* quando receber o pacote. De seguida, o vizinho escolhido, procede de igual forma ao veículo que teve o acidente e assim sucessivamente até chegar ao último que estiver dentro da ZoR. Os veículos selecionados são denominados *MaxRank*, sendo o *rank* inicialmente, nesta estratégia, a distância de cada

vizinho em relação ao acidente e depois, de cada vizinho relativamente ao *MaxRank* escolhido e assim sucessivamente. Uma vez que esta estratégia faz a escolha do próximo *MaxRank* através da maior distância, os veículos que se encontram à frente do *MaxRank* atual, na ZoR, vão receber uma mensagem repetida, pois já a receberam do *MaxRank* anterior. A forma como são identificados os veículos que estão sucessivamente à frente dos *MaxRanks* depende da localização dos veículos e da sua direção do movimento. Os que estiverem fora dessa área ouvem a mensagem, mas vão descartá-la, por se encontrarem fora da área de interesse definida inicialmente (isto é, as suas coordenadas não estão dentro da circunferência delimitada pelas coordenadas do acidente e pelo raio). Os que estiverem na direção contrária à do acidente ou que se encontrem à frente do mesmo também vão descartá-la, pois a sua trajetória não é afetada, e consequentemente a mensagem não é considerada relevante.

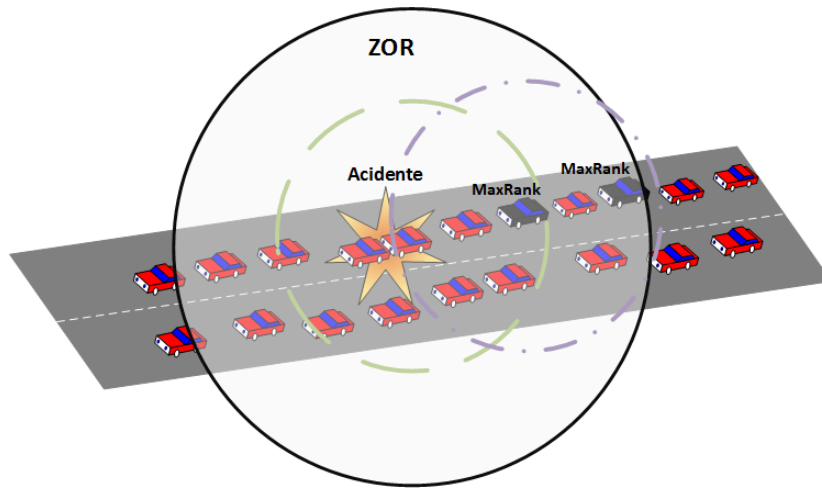


Figura 3.2: Cenário da estratégia de Alta Densidade

Esta seleção estratégica tem como objetivo reduzir o número de retransmissões, não afetando a chegada da informação a todos os veículos na ZoR (taxa de entrega). Além disso, todos os veículos enviam periodicamente mensagens de controlo para os seus vizinhos com as suas informações, nomeadamente a sua localização.

Lógica de Envio de Pacotes de Dados

A figura 3.3(a) representa o diagrama de fluxo do funcionamento da lógica de envio dos pacotes de dados relativamente ao algoritmo de Alta Densidade no emulador utilizado, mOVERS.

Quando a *thread* `runRouting` inicia, é executada a estratégia de Alta Densidade, se o ficheiro de configuração contiver a informação que indica que a versão a ser executada corresponde à de Alta Densidade. Esta *thread* é executada com uma periodicidade predefinida `CYCLE_DELAY` micro segundos (o valor por defeito é 200 ms).

Inicialmente, o nó verifica se o pacote que vai ser enviado é de conteúdo urgente através do tipo de serviço contido no pacote de emergência. O processo de disseminação só começa quando o nó que teve um acidente, por exemplo (pode ser o nó atrás deste), entrar neste método e tiver na sua vizinhança OBUs. É ele quem gera o pacote com as informações relativas à sua localização. De seguida, calcula a próxima OBU que vai ser o primeiro *MaxRank*, de

acordo com o fluxograma do algoritmo 1 (será explicado mais à frente com maior detalhe) e atualiza o pacote com as informações do mesmo, nomeadamente coordenadas GPS e o ID. A OBU que gera o pacote de emergência envia-o (informando, por exemplo, que teve um acidente ou que fez uma travagem brusca) para todos os seus vizinhos e, no final do envio atualiza o módulo de *Logging*. Se as OBUs não forem *MaxRank*, não enviam nenhum pacote. No entanto, se essa condição se verificar, isto é, o ID que estiver no cabeçalho do pacote correspondente ao *MaxRank* for o seu, essa OBU vai calcular o próximo *MaxRank* e atualiza novamente o pacote com essa informação. Depois, envia o pacote para todos os vizinhos já com a nova informação acerca do próximo *MaxRank*. Por fim, é atualizado o módulo de *Logging*.

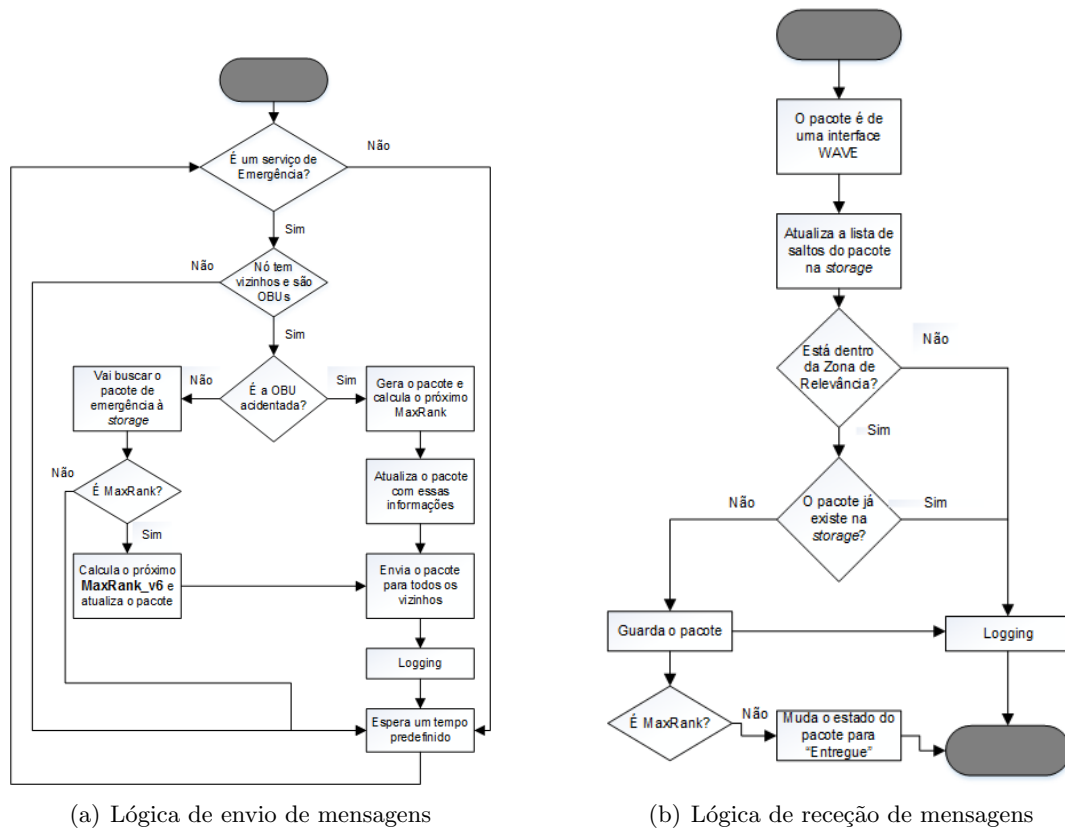


Figura 3.3: Diagramas de fluxo das lógicas dos pacotes de emergência correspondentes da estratégia de Alta Densidade

Lógica de Receção de Pacotes de Dados

A 3.3(b) representa o diagrama de fluxo do funcionamento da lógica de receção dos pacotes de dados relativamente à estratégia de Alta Densidade no mOVERS.

Quando um nó recebe um pacote de emergência vindo do módulo *RX*, inicia-se o processo de receção de um pacote de dados. Se este for recebido através de uma interface *WAVE*, é atualizada a lista de saltos do pacote com o ID do nó que recebeu o pacote. É importante realçar que as OBUs apenas comunicam através das interfaces *WAVE* e, por este motivo o

envio e receção de pacotes é feito através da tecnologia WAVE. De seguida, o nó verifica se está dentro da ZoR, através das suas coordenadas, e se estiver, confirma se já tem o pacote guardado. Após isto, é atualizado o módulo de *Logging*, caso o nó já tenha esse pacote ou caso ainda não o tenha, isto é, é atualizada apenas a variável correspondente a uma destas duas situações. Se a OBU estiver fora da ZoR e se receber um pacote, irá descartá-lo, pelo que o módulo de *Logging* terá de ser novamente atualizado.

Uma vez que os nós ao receberem um pacote verificam primeiro se já o têm guardado na *storage* e se o nó que recebeu o pacote for *MaxRank*, este não irá conseguir assumir-se como tal, porque considera o pacote repetido e, por isso, não verifica mais nenhuma informação contida no mesmo. Desta forma, ignora o campo `nextMaxRankID` contido no pacote. Caso o nó verificasse em primeiro lugar se era ele o próximo *MaxRank*, iria enviar um pacote repetido para a sua vizinhança e possivelmente esta também já o teria recebido. Esta decisão iria originar mais pacotes redundantes na rede, contrariando um dos objetivos desta dissertação que é reduzir a quantidade de pacotes redundantes.

Por fim, apenas as OBUs que não são *MaxRank* mudam o estado do pacote, de forma a sinalizar que este já foi entregue e recebido pelas OBUs e não será reenviado pelas mesmas. Caso sejam *MaxRanks*, quando recebem o pacote guardam-no, mas não alteram o estado do pacote, pois pretendem reenviá-lo para todos os seus vizinhos.

O algoritmo 1 descreve a forma como é feito o cálculo dos nós (*MaxRanks*) escolhidos para reenviar os pacotes de conteúdo urgente em *broadcast*, na ZoR.

Algoritmo 1: Cálculo do *MaxRank* para a estratégia de Alta Densidade

Entrada: Pacote de emergência
Saída: ID do próximo *MaxRank*

```

1  início
2  if É o MaxRank atual then
3      if Tem vizinhos then
4          repita
5              if Está dentro da ZOR then
6                  if Está atrás do MaxRank then
7                      Calcula e guarda a distância em relação ao vizinho;
8                      if Distância atual é maior do que a guardada then
9                          Trata novo MaxRank;
10                         Guarda a distância para comparar com o próximo vizinho;
11                         Guarda o ID do vizinho como MaxRank;
12                     Próximo vizinho;
13                 até Não haver mais vizinhos;
14             Atualiza a estrutura do MaxRank;
15     Devolve o ID do próximo MaxRank;
16 fim

```

De cada vez que é efetuado o cálculo do próximo *MaxRank*, o atual confirma que é ele quem vai fazer essa escolha, isto é, no primeiro caso, a OBU acidentada calcula o próximo *MaxRank*, uma vez que, inicialmente, o pacote contém a informação que é ela o *MaxRank*. De seguida, vai analisar as coordenadas de cada um dos seus vizinhos, que são enviadas nos pacotes de *advertisements* e ver se estão dentro da ZoR, através de cálculos (verifica se os dois pontos das

coordenadas dos vizinhos pertencem à circunferência de raio igual ao comprimento da ZoR e centrada nas coordenadas da OBU que teve o acidente). Caso não esteja, o procedimento termina. Se estiver, vai verificar se o vizinho está atrás de si, pois a escolha será feita apenas sobre os vizinhos que estiverem atrás do *MaxRank*. Esta escolha envolve uma verificação da direção da OBU que teve o acidente, para delimitar o ângulo que vai servir, de seguida, para fazer a comparação entre as direções de deslocamento dos vizinhos e do acidente (ou do *MaxRank*), de forma a verificar se estes estão dentro dos limites de direção previamente definidos. Após estas verificações vão ser analisadas e comparadas as coordenadas dos vizinhos e do *MaxRank*, para determinar se estão atrás do mesmo, dependendo do valor da direção do acidente e do quadrante em que se encontre na circunferência que define a direção do movimento (Figura 3.4). Se estas condições não se verificarem, o processo termina. Caso contrário, vai calcular a sua distância relativamente a cada vizinho, guardando sempre a maior para ir comparando sucessivamente com aquela que está a ser calculada no momento, bem como o ID do vizinho a que corresponde essa maior distância. Por fim, quando encontrar o vizinho que tem a maior distância, atualiza a estrutura com as informações do novo *MaxRank* e devolve o ID do mesmo. Esse novo ID será atualizado no cabeçalho do pacote.

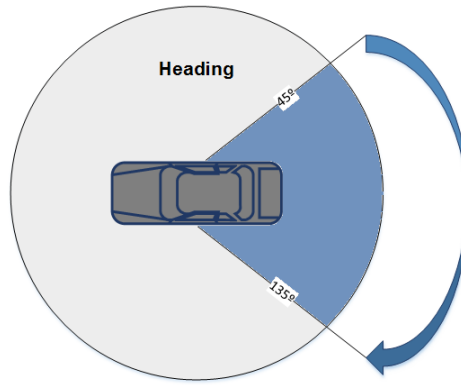


Figura 3.4: Direção da Disseminação do movimento

Lógica de Envio de Pacotes de *Advertisement*

O algoritmo 2 representa o modo de funcionamento do envio de pacotes de *advertisement* na rede por todos os nós.

O envio de pacotes de *advertisement* para todos os vizinhos é iniciado quando a *flag* que está responsável por indicar o envio desses pacotes estiver ativa. Neste trabalho, esta *flag* está ativa apenas para as OBUs e é ativada previamente. Neste caso, apenas as OBUs têm esta *flag* ativa e, por isso, sempre que têm vizinhos trocam este tipo de pacotes (*advertisement*) entre si. Cada nó envia para todos os seu vizinhos pacotes que contêm a latitude, longitude e direção do movimento do próprio nó. No final de cada envio, é atualizado o módulo de *Logging*. O pacote de controlo será enviado novamente após esperar um tempo aleatório entre dois valores (mínimo e máximo) previamente definidos.

Algoritmo 2: Lógica de envio de *advertisements* para estratégia de Alta Densidade

Saída: Pacote de *advertisement*

```
1 início
2   repita
3     Flag is2sendADV ativa;
4     Criação do pacote de advertisement com as coordenadas atuais do nó;
5     Envio do pacote de advertisement para todos os vizinhos;
6     Logging;
7     Espera um tempo aleatório entre MIN_ADV_PER e MAX_ADV_PER;
8   até Fim da ZOR;
9 fim
```

Lógica de Receção de Pacotes de *Advertisement*

O algoritmo 3 representa o modo de funcionamento da receção de pacotes de *advertisements* na rede por todos os nós.

Quando um nó recebe um pacote vindo de uma interface WAVE, verifica na sua lista de vizinhos, se esse pacote pertence a um nó que é seu vizinho. Se for, atualiza as informações relativas a esse vizinho, nomeadamente as coordenadas GPS, na estrutura de vizinhos. Se não for, altera uma *flag* auxiliar que indica que já não é mais seu vizinho e elimina-o da sua lista de vizinhos.

Algoritmo 3: Lógica de receção de *advertisements* para estratégia de Alta Densidade

Entrada: Pacote de *advertisement*

```
1 início
2   if O pacote é de uma interface WAVE then
3     if É meu vizinho then
4       | Atualiza as informações do vizinho;
5     else
6       | Altera a flag que indica que já não é vizinho;
7       | Elimina o vizinho da sua lista;
8     end
9   else
10    | Descarta o pacote;
11  end
12 fim
```

3.3.2 Estratégia de Baixa Densidade

Esta estratégia, ao contrário da anterior, foi feita tendo em consideração cenários de baixa densidade, isto é, situações onde poderão existir nós isolados ou com poucos vizinhos. O método de seleção do *MaxRank* depende diretamente do número de vizinhos que cada veículo tem atrás. É desta forma que é atribuído um *rank* a cada veículo. Inicialmente, o veículo que teve um situação de emergência, por exemplo, um acidente analisa o número de vizinhos que cada veículo tem atrás e escolhe o que tem mais vizinhos atrás para retransmitir a mensagem

(*MaxRank*). Esta informação relativa à quantidade de vizinhos que cada vizinho tem atrás é enviada por cada nó nos *advertisements*. O próximo *MaxRank* procede da mesma maneira com todos os seus vizinhos e este processo repete-se até ao último veículo contido na ZoR, ou seja, cujas coordenadas estejam muito perto do limite ou no limite da circunferência que define a ZoR. Uma vez que é um cenário de baixa densidade, é mais vantajoso escolher um veículo que tenha mais vizinhos atrás de si, isto é, com maior *rank*, pois consegue disseminar o pacote de emergência para mais veículos com o envio de apenas uma mensagem para todos os seus vizinhos.

Este critério de seleção também reduz o número de retransmissões na rede, e todos os veículos enviam pacotes de controlo periodicamente para os seus vizinhos, de forma a terem conhecimento da rede, tal como a estratégia descrita anteriormente.

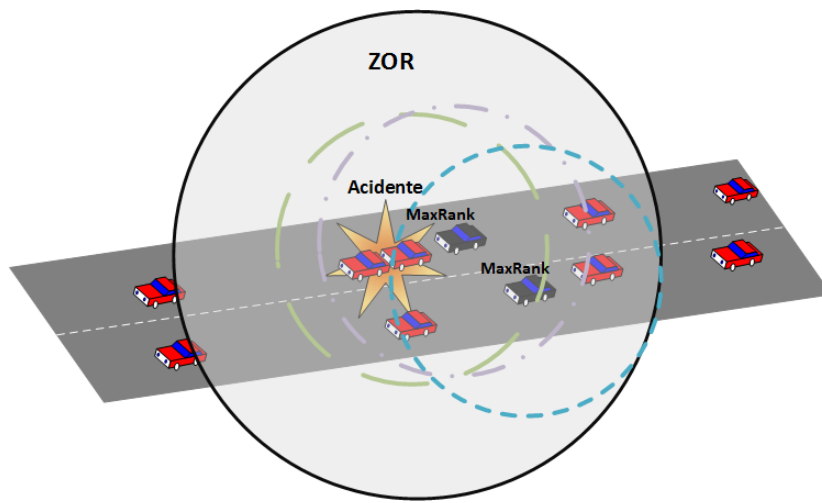


Figura 3.5: Cenário da estratégia de Baixa Densidade

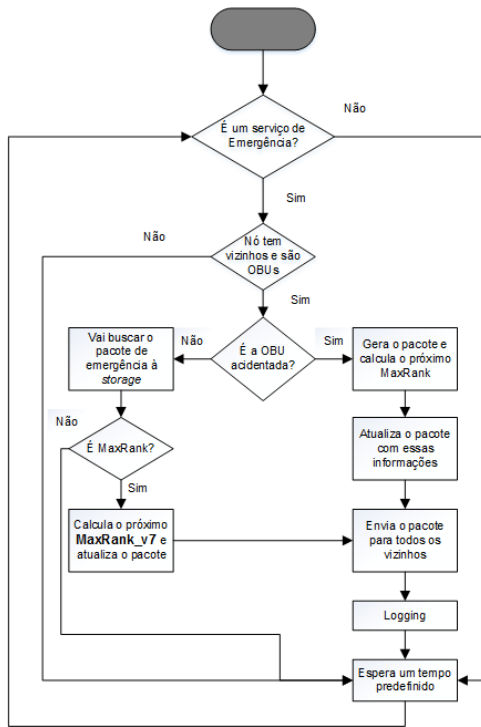
Lógica de Envio de Pacotes de Dados

A figura 3.6(a) representa o diagrama de fluxo do funcionamento da lógica de receção relativamente à estratégia de baixa densidade no mOVERS.

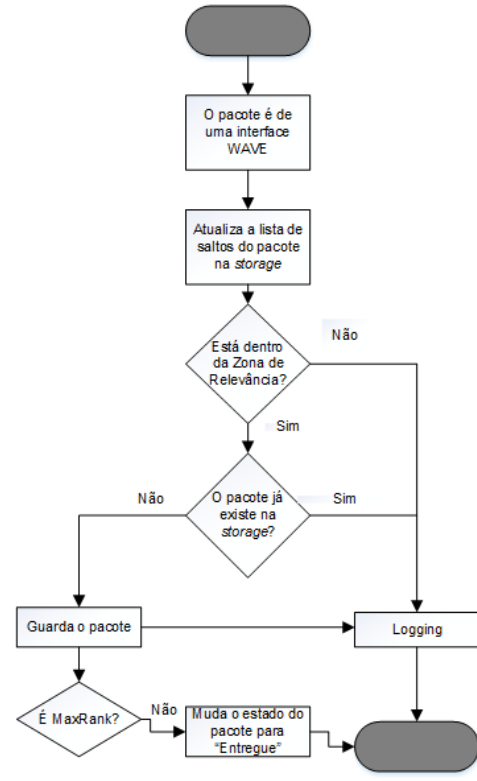
Esta estratégia de envio de pacotes de dados é muito semelhante à descrita para a estratégia de Alta Densidade. No entanto, o método de seleção do *MaxRank* é diferente e está descrito no algoritmo 4.

De cada vez que é efetuado o cálculo do próximo *MaxRank*, o atual confirma que é ele quem vai fazer essa escolha. No primeiro caso, a OBU acidentada verifica através da informação contida no pacote de emergência que o seu ID corresponde ao *MaxRank* atual. De seguida, vai analisar as coordenadas de cada um dos seus vizinhos e ver se estão dentro da ZoR, através de cálculos (verifica se os dois pontos das coordenadas dos vizinhos pertencem à circunferência de raio igual ao comprimento da ZoR e centrada nas coordenadas da OBU que teve o acidente). Estas coordenadas serão usadas mais à frente no processo para tomar a decisão final da escolha do *MaxRank*, caso aconteça um empate, ou seja, igual número de vizinhos atrás. Caso não esteja atrás do *MaxRank*, o procedimento termina. Se estiver, vai verificar se o vizinho está atrás de si, pois a escolha será feita apenas sobre os vizinhos que

estiverem atrás do *MaxRank*. A forma como é feita esta verificação é igual à da estratégia de Alta Densidade. A partir daqui, inicia o processo de seleção guardando, sucessivamente a quantidade de vizinhos atrás (que é enviada nos pacotes de controlo) que cada vizinho seu tem, para ir comparando essas quantidades e guardar o maior valor. Sempre que encontra um vizinho com uma quantidade de vizinhos atrás maior do que aquela que tem guardado, guarda esse valor, o ID do respetivo vizinho e as coordenadas GPS relativas ao mesmo. Em caso de empate, o vizinho escolhido será o que estiver mais longe, e é guardado novamente o ID, que corresponde ao novo *MaxRank*. Por fim é atualizada a estrutura do *MaxRank* com as novas informações e devolvido o ID do mesmo. Este ID é atualizado no pacote de emergência que será enviado pelo *MaxRank*.



(a) Lógica de envio no mOVERS



(b) Lógica de receção no mOVERS

Figura 3.6: Diagramas de fluxo das lógicas dos pacotes de emergência correspondentes à estratégia de Baixa Densidade

Algoritmo 4: Cálculo do *MaxRank* para a estratégia de Baixa Densidade

Entrada: Pacote de emergência**Saída:** ID do próximo MaxRank

```
1 início
2   if É o MaxRank atual then
3     if Tem vizinhos then
4       repita
5         if Está dentro da ZOR then
6           if Está atrás do MaxRank then
7             if É o primeiro vizinho then
8               Guarda as coodenadas do vizinho e o número de vizinhos que
               este tem atrás;
9             else
10              if Número de vizinhos é maior ou igual que o anterior then
11                if Tem o mesmo número de vizinhos then
12                  Guarda e calcula as distâncias entre o MaxRank e
                  esses vizinhos;
13                  if Distância é menor que a anterior then
14                    Próximo vizinho;
15                  Guarda as coordenadas GPS e o número de vizinhos;
16                  Guarda o ID do vizinho como MaxRank;
17                end
18              Próximo vizinho;
19            até Não haver mais vizinhos;
20          Atualiza a estrutura do MaxRank;
21        Devolve o ID do próximo MaxRank;
22 fim
```

Lógica de Receção de Pacotes de Dados

A lógica de receção dos pacotes de dados (emergência) é igual à da estratégia de Alta Densidade, uma vez que o objetivo é o mesmo, disseminar o pacote de emergência por todas as OBUs que se encontram atrás do acidente e dentro da ZoR.

Lógica de Envio de Pacotes de *Advertisement*

A figura 3.7(a) representa o modo de funcionamento do envio de pacotes de *advertisements* na rede por todos os nós.

O processo de envio de pacotes de controlo inicia quando a *flag* responsável por indicar o início do envio destes pacotes estiver ativa. Neste caso, apenas as OBUs terão essa *flag* ativa, tal como acontece na estratégia de Alta Densidade, uma vez que nesta dissertação apenas foram consideradas OBUs. Cada nó cria um pacote com as suas coordenadas atuais (latitude, longitude e direção) e depois verifica quantos vizinhos tem atrás de si e guarda esse valor. De seguida, envia o pacote de controlo com essas informações para todos os seus vizinhos. Esta informação vai ser útil quando for necessário calcular o *MaxRank* na lógica de envio

de pacotes de dados explicada anteriormente. Por fim, o módulo de **Logging** é atualizado. O pacote de controlo será novamente enviado, após esperar um tempo aleatório entre dois valores (mínimo e máximo) previamente definidos.

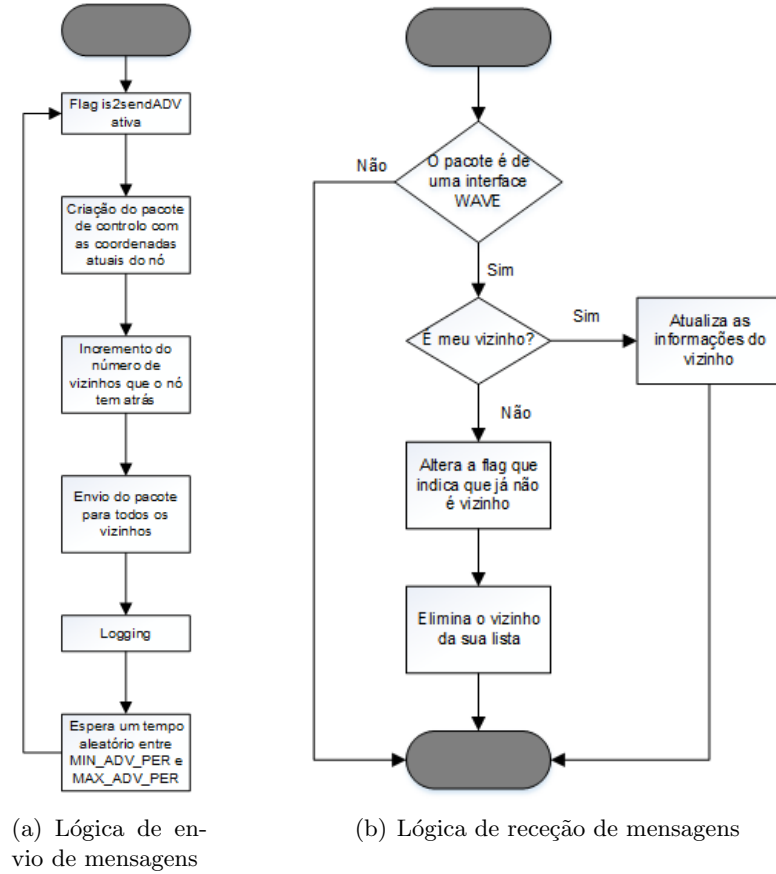


Figura 3.7: Diagramas de fluxo das lógicas dos pacotes de *advertisements* correspondentes à estratégia de Baixa Densidade

Lógica de Receção de Pacotes de *Advertisement*

A figura 3.7(b) apresenta o modo de funcionamento da receção de pacotes de *advertisements* na rede por todos os nós.

Relativamente à receção dos pacotes de controlo, o processo é igual ao descrito na estratégia de Alta Densidade. Pelo que, quando recebe um pacote deste tipo, atualiza a sua lista de vizinhos e a estrutura de vizinhos com as novas informações ou elimina o vizinho da mesma e atualiza a lista de vizinhos.

3.3.3 Estratégia de *Broadcast*

Esta estratégia serve de base para comparar com as outras duas descritas anteriormente, pois não foi pensado para cenários com uma densidade específica. A partir do momento em que se dá o acidente, todos os veículos contidos na ZoR enviam sucessivamente uma mensagem

para todos os seus vizinhos, isto é, não existe o conceito de *MaxRank*. Os veículos não necessitam de ter um conhecimento da rede e apenas precisam de conhecer a sua localização para determinarem se estão dentro ou fora da ZoR e, conseqüentemente, perceberem se devem ou não descartar as mensagens recebidas.

Lógica de Envio de Pacotes de Dados

A figura 3.8(a) apresenta o diagrama de fluxo do funcionamento da lógica de envio de pacotes de dados na estratégia de *Broadcast*.

Quando a *thread runRouting* inicia, é executada a estratégia de *Broadcast*, caso o ficheiro de configuração contenha a informação que indique que a estratégia selecionada é a de *Broadcast*. Esta *thread* é executada com uma periodicidade predefinida *CYCLE_DELAY*, em micro segundos (o valor por defeito é 200 ms).

Inicialmente, o nó verifica se o pacote que vai ser enviado é de conteúdo urgente. O processo de disseminação só começa quando o nó que teve o acidente entrar neste método e tiver na sua vizinhança OBUs. É ele quem gera o pacote de dados com as informações relativas à sua localização e envia o mesmo para todos os seus vizinhos. Depois de enviar o pacote, altera o estado do mesmo para indicar que este foi entregue e não voltar a enviá-lo. A partir daqui, as OBUs que receberam o pacote vão à *storage* buscar um pacote de emergência e, se estiverem dentro da ZoR enviam o pacote para todos os seus vizinhos. De seguida, alteram o estado do pacote para indicar que este foi entregue e não voltar a enviá-lo. Por fim, as variáveis correspondentes aos pacotes recebidos, repetidos e descartados do módulo de *Logging* são atualizadas.

Lógica de Receção de Pacotes de Dados

A figura 3.8(b) representa o diagrama de fluxo do funcionamento da lógica de receção de pacotes de dados na estratégia de *Broadcast*.

A lógica de receção de pacotes de dados é muito semelhante à das outras duas estratégias. Quando um nó recebe um pacote numa interface WAVE, atualiza a lista de saltos do mesmo na *storage*. Se estiver dentro da ZoR e se ainda não tiver recebido aquele pacote, guarda-o na *storage*. De seguida, o módulo de *Logging* é atualizado. Se já tiver esse pacote, significa que é repetido e, portanto, a variável correspondente aos pacotes repetidos é atualizada no *Logging*. Se a OBU estiver fora da ZoR e receber o pacote de emergência, é atualizada a variável correspondente aos pacotes descartados, no *Logging*.

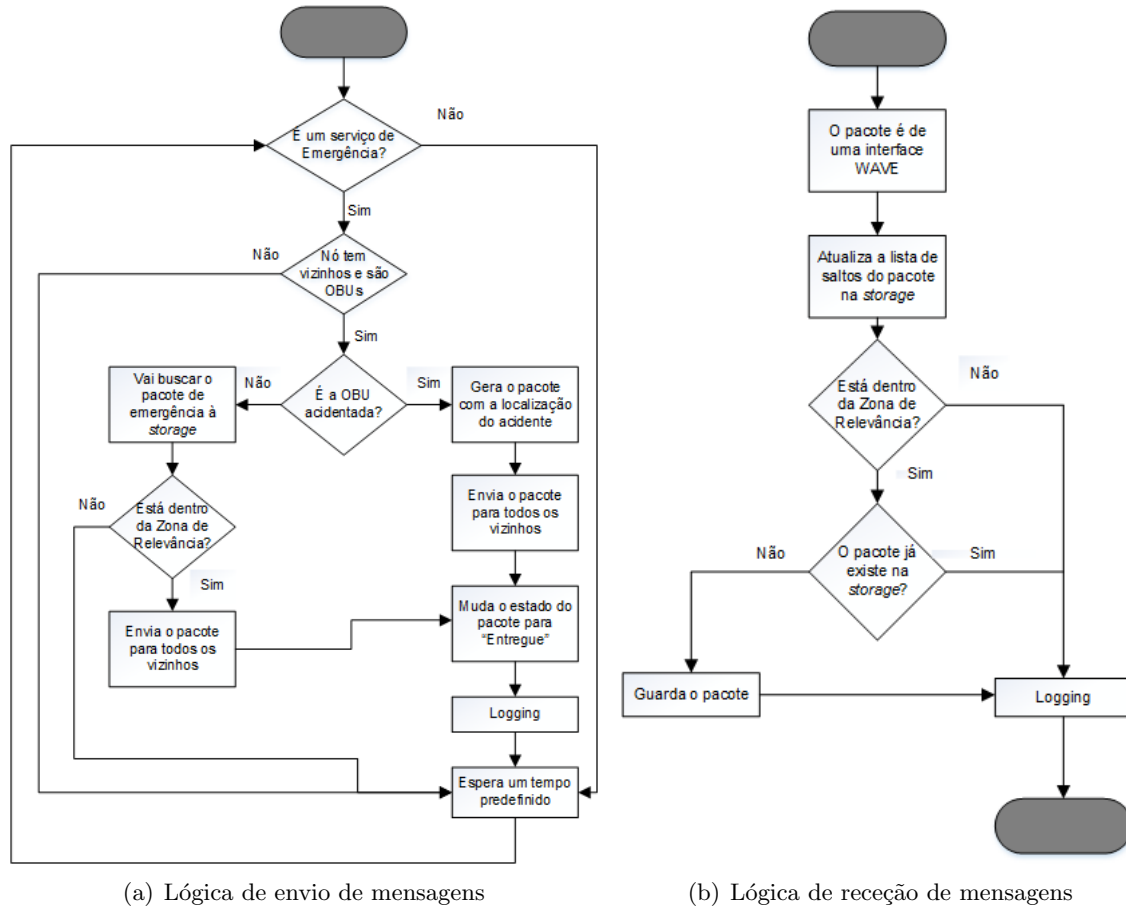


Figura 3.8: Diagramas de fluxo das lógicas dos pacotes de emergência correspondentes à estratégia de *Broadcast*

3.4 Considerações do Capítulo

As duas estratégias existentes para os dois tipos de densidade (alta e baixa) têm a vantagem de reduzir o número de retransmissões e a quantidade de pacotes de dados repetidos que circulam na rede. Portanto, o seu objetivo é que a mensagem chegue a todos os veículos que estão na ZoR recorrendo a muito menos transmissões do que na estratégia em que todos enviam em *broadcast*, isto é, para todos os seus vizinhos.

No próximo capítulo será apresentada a arquitetura e o funcionamento do emulador utilizado para implementar e testar as estratégias de disseminação de mensagens de emergência propostas. Serão apresentados e descritos os módulos que constituem o emulador usado, mOVERS, as funções de cada módulo e a forma como é feita a recolha de dados. Finalmente serão apresentadas as modificações realizadas nos módulos para a implementação das estratégias de disseminação de mensagens de emergência propostas.

Capítulo 4

Implementação e Integração

4.1 Descrição do Capítulo

No capítulo anterior foram apresentadas e descritas as três estratégias desenvolvidas e implementadas no mOVERS: Alta e Baixa Densidade e *Broadcast*. Para tal, foram usados algoritmos e diagramas de fluxo, de forma a facilitar a compreensão das mesmas e as lógicas subjacentes a cada uma. Foi também explicado o problema que deu origem a esta dissertação.

Neste capítulo será apresentada a arquitetura e o funcionamento do emulador (mOVERS) utilizado para avaliar as estratégias implementadas. Serão descritos todos os módulos existentes no mOVERS, bem como as suas funções e a forma como é feita a recolha de dados da Base de Dados usada neste trabalho (cujos dados foram gerados aleatoriamente). Por fim, serão apresentadas todas as modificações feitas nos módulos, de forma a ser possível implementar as três estratégias e as restrições existentes no emulador para a implementação das estratégias de disseminação de mensagens de emergência.

4.2 Descrição do mOVERS

As estratégias de encaminhamento desenvolvidas nesta dissertação, foram implementadas e testadas na plataforma mOVERS. O mOVERS surgiu através de uma parceria entre a Veniam[®] e o grupo de investigação NAP do IT da Universidade de Aveiro. Este emulador tem por base uma DTN e foi desenvolvido com o intuito de poder emular uma rede veicular em que cada OBU e RSU têm o mesmo código de funcionamento que os elementos reais. Suporta comunicações entre veículos e diversos protocolos, nomeadamente WAVE, (IEEE802.11p) e Wi-Fi (IEEE802.11a/b/g). A escolha desta plataforma para a implementação das estratégias prende-se com o facto do código desenvolvido no mOVERS ter a particularidade de ser igualmente aplicável tanto nos veículos emulados como OBUs e RSUs, como nas placas que se encontram na rede real no Porto.

4.2.1 Visão Geral da Arquitetura e Funcionamento

O mOVERS permite criar um ambiente emulado para redes veiculares, de forma a ser possível implementar e avaliar estratégias de disseminação de conteúdos (urgentes e não urgentes), usando os mecanismos da DTN. Além disso, permite criar vários processos (cada um correspondente a uma OBU ou RSU, por exemplo) que correm na DTN. Estes processos que

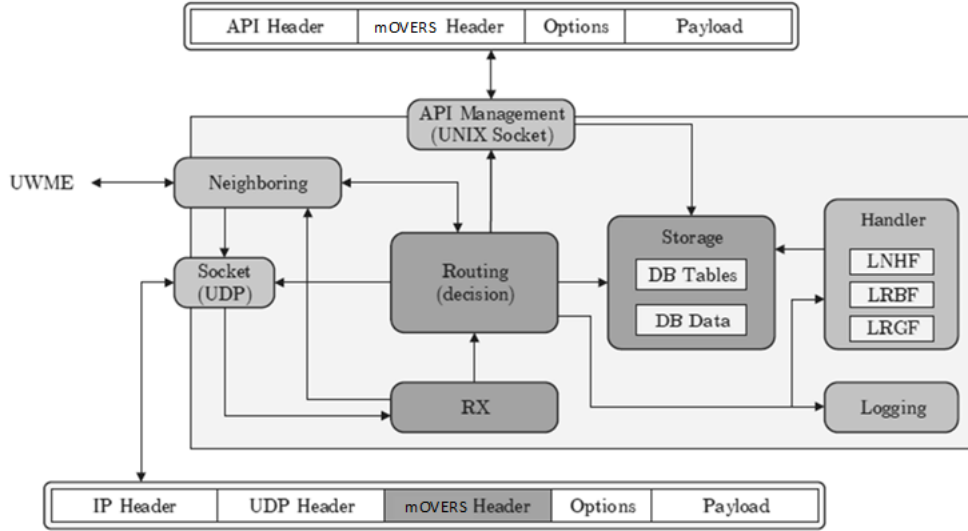


Figura 4.1: Arquitetura do mOVERS [27]

correspondem a nós emulados da rede, são obtidos através da recolha de dados reais vindos da rede veicular do Porto, ou também podem ser obtidos através de uma base de dados gerada com dados aleatórios (como é o caso desta dissertação).

A arquitetura do mOVERS está presente na Figura 4.1 e é composta por diversos módulos que serão descritos a partir da Subsecção 4.2.3. Os módulos interagem entre si através dos *Inter-Process Communication (IPC) sockets*, e um nó pode trocar pacotes com outros nós, usando o protocolo *User Datagram Protocol (UDP)*. Existem diversos tipos de nós vizinhos, consoante a interface que está a ser usada (Wi-Fi, WAVE e *Ethernet*). Uma vez que o mOVERS foi desenvolvido para redes veiculares, os vizinhos WAVE considerados são OBUs ou RSUs, os nós que contêm interfaces Wi-Fi são sensores e as RSU e servidores localizados na rede principal estão conectados a uma ligação estática ou uma interface *Ethernet*.

De forma ao uso do emulador poder ser escalável em redes veiculares, foram criadas as mensagens *Zero-M Queue (ZMQ)* [48], para conseguir lidar com um número elevado de processos (cada processo corresponde a um veículo na rede).

Assim, existem diversos tipos de mensagens que são trocadas entre o mOVERS e a DTN, tais como as mensagens de *advertisement* enviadas pelo emulador e os pacotes regulares trocados entre nós (dados e *advertisement*). Portanto, quando um nó recebe uma mensagem de *advertisement*, as informações mais importantes como a posição do nó e a sua lista de vizinhos são atualizadas, mas o módulo *Socket* não a recebe, pois não é um pacote regular. Se for recebido um pacote regular, este é encaminhado para o módulo *Socket* e, posteriormente, consoante a classificação feita no módulo *RX*, o pacote será novamente encaminhado para um dos seguintes módulos: *Routing* ou *Neighboring*.

De forma a ser possível trocar mensagens entre os processos que estão a ser emulados, foi criada a classe `EmuMessageHeader` que define o cabeçalho das mensagens (Figura 4.2).

Assim, quando um nó envia informação para um ou mais nós, é criado um objeto desta classe e os campos são inicializados. Esse objeto constitui um cabeçalho unido ao pacote que vai ser enviado e é utilizado pelo emulador para o encaminhar até ao seu destino.

O mOVERS permite seleccionar os veículos que vão ser emulados, não necessitando de

EmuMessageHeader
+ emu_key : uint32_t
+ msg_type : uint32_t
+ if_type : uint32_t
+ dest_node : DTNeid
+ src_node : DTNeid
+ user_data_length : size_t
+ print() : void

Figura 4.2: Classe do cabeçalho das mensagens do mOVERS [49]

emular todos os veículos que se encontram na rede real. Desta forma, o utilizador selecciona os veículos de acordo com os seus IDs ou com a sua posição geográfica, isto é, poderão estar contidos numa região de interesse à disseminação de conteúdos (uma área urbana, por exemplo) na rede.

4.2.2 Recolha de Dados da Base de Dados MySQL

O emulador envia mensagens de *advertisement* periodicamente para todos os nós e, desta forma, eles vão atualizando o seu estado. Essas informações estão armazenadas numa base de dados e vão sendo efetuados pedidos à mesma. Esses dados são reais e foram recolhidos durante um dia da rede veicular real existente no Porto.

O mOVERS organiza a informação em três tabelas: "*timestamps*", "*neighbors*" e "*rsu*", cujos campos estão detalhados na Tabela 4.1, Tabela 4.2 e Tabela 4.3. Nesta dissertação apenas foram usadas duas das três tabelas referidas (4.1 e 4.2). A Tabela 4.1 contém informação relativa a cada nó, em cada *timestamp*. A Tabela 4.2 está relacionada com a primeira, isto é, corresponde aos IDs dos vizinhos e o respetivo *RSSI*, que o nó da Tabela 4.1 tem contacto, nesse instante temporal. A Tabela 4.3 refere-se à informação que cada RSU tem nos instantes em que os dados foram recolhidos na rede real durante um dia (24h).

Entry ID	UNIX Timestamp	Node ID	GPS Coordinates	Direction	Speed	Cell	Cell RSSI
----------	----------------	---------	-----------------	-----------	-------	------	-----------

Tabela 4.1: Base de Dados do mOVERS: Tabela por *Timestamps* [49]

Entry ID	UNIX Timestamp	Neigh ID	Cell RSSI
----------	----------------	----------	-----------

Tabela 4.2: Base de Dados do mOVERS: Tabela por Vizinhos [49]

No entanto, neste trabalho a base de dados utilizada não contém dados recolhidos da rede real, mas sim dados gerados aleatoriamente, de acordo com os campos pertencentes às tabelas 4.1 e 4.2.

RSU ID	Location	mOVERS Version	Board Version	Owner	VNetwork	provider	uptime	Last Ping	GPS Coordinates
--------	----------	----------------	---------------	-------	----------	----------	--------	-----------	-----------------

Tabela 4.3: Base de Dados do mOVERS : Tabela por RSUs [49]

Estes dados foram gerados para conseguir criar diferentes cenários controlados e poder testar as estratégias aqui propostas, foi alterada a base de dados existente para uma adequada aos cenários de estudo, com um determinado número de veículos e informações não reais sobre os mesmos.

As mensagens de *advertisement* que são enviadas pelo *mOVERS*, estão implementadas numa estrutura designada *EmuControlMsg* e contêm informações importantes acerca de cada nó, tais como a latitude e longitude. Para conseguir ter mais precisão nas coordenadas, foi alterado o tipo destas duas variáveis de *float* para *double*, como se pode ver na Figura 4.3.

EmuControlMsg
+ latitude: double + longitude: double + heading: int + vel: int + cell_rssi: int + connection_type: emu_connection + num_neigh: uint8_t + neigh_id: DTNeid[20] + neigh_rssi: uint8_t[20] + neigh_isRSU: bool[20]

Figura 4.3: Classe das mensagens de *advertisement* do mOVERS [49]

De forma a ser possível o emulador utilizar os dados relativos a cada nó (OBU), é necessário importá-los para uma base de dados através de um *script*. No entanto, antes de importar os dados, deve-se ter em atenção às especificações do *script*. Existem duas Application Programming Interfaces (APIs) disponíveis, uma vai buscar a lista de RSUs e a outra devolve o estado da rede em cada *timestamp*.

4.2.3 Descrição dos módulos do mOVERS

Módulo API

Os nós do mOVERS conseguem interagir com aplicações externas através deste módulo. Este usa mensagens IPC com o objetivo de separar as mensagens de *advertisements* e de dados.

Este módulo trata os pacotes do mOVERS através do API *socket* e cria uma camada de abstração para trocar pacotes entre o mOVERS e o API. Além disso, gere o acesso das aplicações ao mOVERS.

Módulos de Comunicação

Os módulos **Socket** e **RX** são responsáveis pela troca de pacotes de dados e de *advertisements* entre os nós da rede.

O módulo **Socket** gere o acesso ao UDP **Socket** sempre que um vizinho envia um pacote para um nó vizinho ou recebe um pacote vindo do mesmo.

O módulo **RX** classifica e encaminha os pacotes que chegam através do UDP **Socket**, analisando alguns campos do pacote tais como: o destino (**dstEID**), a *flag* (**DTN_FLAG_NEIGH_ACK_MASK**, **DTN_FLAG_END_ACK_MASK** e **DTN_FLAG_ADV_MASK**) e o tipo de serviço (**serviceID**). Após esta análise, o pacote é encaminhado para o módulo de **Routing**, se for um pacote de dados, ou para o módulo de **Neighboring**, caso seja um pacote de anúncio de vizinhos e atualiza a informação relativa à vizinhança, nesse mesmo módulo.

Módulo de Routing

Este módulo é responsável por tomar algumas decisões acerca dos pacotes, nomeadamente quais deles serão enviados e para quais vizinhos, e a lógica de envio e receção dos pacotes que está contida em cada estratégia. O mOVERS utiliza uma solução híbrida de *routing*, pois este módulo encaminha os pacotes consoante o tipo de vizinhos (OBU ou RSU) e o tipo de pacote (dados ou *advertisement*). A primeira decisão a ser tomada depende do tipo de pacote e o resto do processo é feito através do tipo de nó. Portanto, de cada vez que um nó quer enviar um pacote, verifica em primeiro lugar se tem vizinhos ao seu alcance, seleciona aqueles para os quais deve enviar e vai à *storage* buscar os pacotes, iniciando o processo de envio dos mesmos. Esse envio depende do tipo de vizinho e, por isso, existem 4 classes de *routing* que foram implementadas tendo isso em conta, como se pode ver pela Figura 4.4.

As estratégias desenvolvidas neste trabalho para a disseminação de conteúdos urgentes foram incorporadas neste módulo. No entanto, como só vão ser usadas OBUs nesta dissertação, apenas vai ser considerada a classe **RoutingOBU**. A **RoutingOBU** contém duas lógicas: envio e receção de pacotes de dados e *advertisement*, onde estão implementadas as estratégias de disseminação de mensagens de emergência.

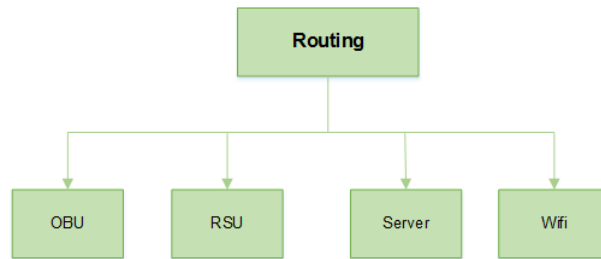


Figura 4.4: Classes dos vários tipos de Routing [27]

Módulo de Logging

Este módulo foi implementado com o objetivo de recolher dados e informações necessárias para avaliar o desempenho das estratégias de disseminação de conteúdos urgentes e não urgentes desenvolvidas no mOVERS. Ao longo de cada experiência, correspondente a uma estratégia de disseminação, é gerada uma pasta com vários ficheiros, onde cada ficheiro corresponde a uma OBU ou RSU e a informação é atualizada em cada **timestamp**. Os dados

guardados para a avaliação do desempenho das várias estratégias de distribuição de conteúdos foram os seguintes:

- `node_eid`: identifica o nó onde está a ser gerada a informação;
- `timestamp`: identifica o instante temporal em que cada iteração foi executada;
- `packets_stored_total`: número total de pacotes guardados por um cada nó ao longo da experiência;
- `packets_transmitted_total`: número total de pacotes transmitidos por cada nó ao longo da experiência;
- `packets_stored_per_timestamp`: número de pacotes guardados em cada `timestamp` (neste caso, de 2 em 2 segundos);
- `packets_transmitted_per_timestamp`: número de pacotes transmitidos em cada `timestamp`;
- `packets_listened_per_timestamp`: número de pacotes "ouvidos" em cada `timestamp`;
- `packets_rcved_good_per_timestamp`: número de pacotes novos guardados, por cada nó em cada `timestamp`;
- `packets_rcved_bad_per_timestamp`: número de pacotes já conhecidos e existentes na *storage* guardados, por cada nó, em cada `timestamp`;
- `control_packets_number_per_timestamp`: número de pacotes de *advertisements* enviados em cada `timestamp`;
- `control_packets_size_per_timestamp`: tamanho total do pacotes de *advertisements* enviados em cada `timestamp`.

Posteriormente foram adicionados mais informações úteis que contribuíram para as métricas de desempenho das estratégias de disseminação de conteúdos urgentes. Essas informações estão descritas pormenorizadamente na Secção 4.3.

Módulo de Neighboring

Como foi referido anteriormente no módulo de comunicação, é através do módulo de **Neighboring** que os nós encontram e guardam os vizinhos na sua lista de vizinhos, quer estes sejam OBUs ou RSUs. Cada nó envia periodicamente pacotes de anúncio, com o objetivo de anunciar a sua presença na rede e para que os nós possam atualizar as suas tabelas com as informações dos seus vizinhos em cada instante.

Como é possível ver na Figura 4.5, este módulo contempla diversos tipos de vizinhos (WiFi, WAVE, por exemplo), dependendo da interface que estiver a ser usada pelos nós para comunicarem entre si. Uma vez que se trata de uma Rede Veicular, cada tipo de nós utiliza uma tecnologia específica para comunicar, sendo que as OBUs e RSUs comunicam através da tecnologia WAVE, os sensores ou *endpoints* utilizam as interfaces Wi-Fi para comunicar e os servidores e as RSU que se encontram no *core* da rede comunicam pela interface *Ethernet*.

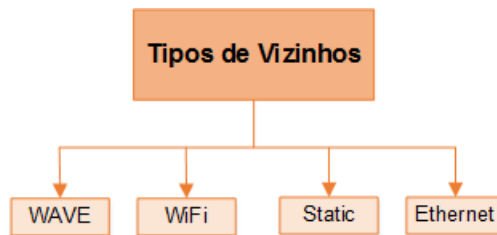


Figura 4.5: Classes dos vários tipos de vizinhos [27]

Módulo de Storage

O módulo de **Storage** é o responsável por guardar vários tipos de pacotes que são importantes para o processo de encaminhamento da informação.

Como se vê na Figura 4.6, este módulo oferece dois tipos de armazenamento: **StorageDisk** e **StorageRAM** que, por sua vez, estão organizados da seguinte forma: **StorageInfoTable** apenas existe na **StorageRAM** e **StorageData** está contida nos dois sub-módulos (**StorageDisk** e **StorageRAM**).

O sub-módulo Random-Access Memory (RAM) engloba 5 tipos de tabelas que permitem um acesso rápido e uma pesquisa otimizada dos pacotes, e cada uma armazena os mesmos de acordo com critérios específicos:

- **Expiry Table:** os pacotes estão ordenados por ordem crescente de tempo de expiração (tempo de vida) do pacote;
- **Hash Table:** os pacotes estão armazenados pelo identificador único de cada pacote (*hash*);
- **NoData Table:** contém os pacotes conhecidos na rede, mas não tem dados guardados. Estes pacotes estão ordenados pelo seu tempo de expiração (tempo de vida);
- **OnHold Table:** os pacotes estão ordenados por ordem crescente do tempo de espera até serem reenviados;
- **Own Table:** os pacotes estão armazenados pelo tipo de serviço (**serviceID**) do mesmo e cujo destino é o próprio nó.

O outro sub-módulo, **StorageDATA** gere o armazenamento persistente dos pacotes no disco, lê e escreve ficheiros (um por cada pacote) para obter o conteúdo dos pacotes, e utiliza as tabelas do sub-módulo **StorageRAM** para melhorar o desempenho dessas operações.

Os pacotes que vão ser encaminhados encontram-se na **Expiry Table**, e o primeiro a ser enviado é o que tiver o menor tempo de vida. Por outro lado, os pacotes que estiverem na **OnHold Table** ficam em espera um tempo compreendido entre um tempo mínimo e máximo de espera (por exemplo entre 60 e 120 segundos). Após este tempo, os pacotes ficam novamente disponíveis para serem reenviados, evitando que estes sejam constantemente enviados quando não é necessário.

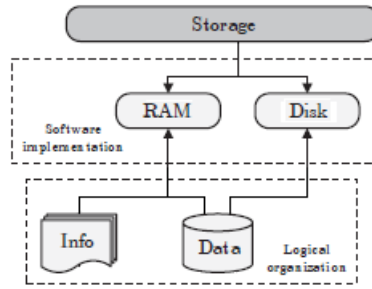


Figura 4.6: Organização do módulo Storage [27]

4.3 Integração e Implementação das Estratégias de disseminação de mensagens de emergência no mOVERS

Nesta secção será descrita a implementação das estratégias no mOVERS, bem como as modificações gerais que foram feitas no mesmo.

4.3.1 Modificações Gerais

Partindo da arquitectura existente (Figura 4.1), foram feitas algumas modificações nos módulos de *Routing*, *Storage*, e *Logging*. Neste último módulo, as alterações foram apenas o tipo de dados que foi recolhido para tirar os resultados pretendidos. Quanto ao *Routing* foram acrescentadas três estratégias de disseminação de conteúdos urgentes.

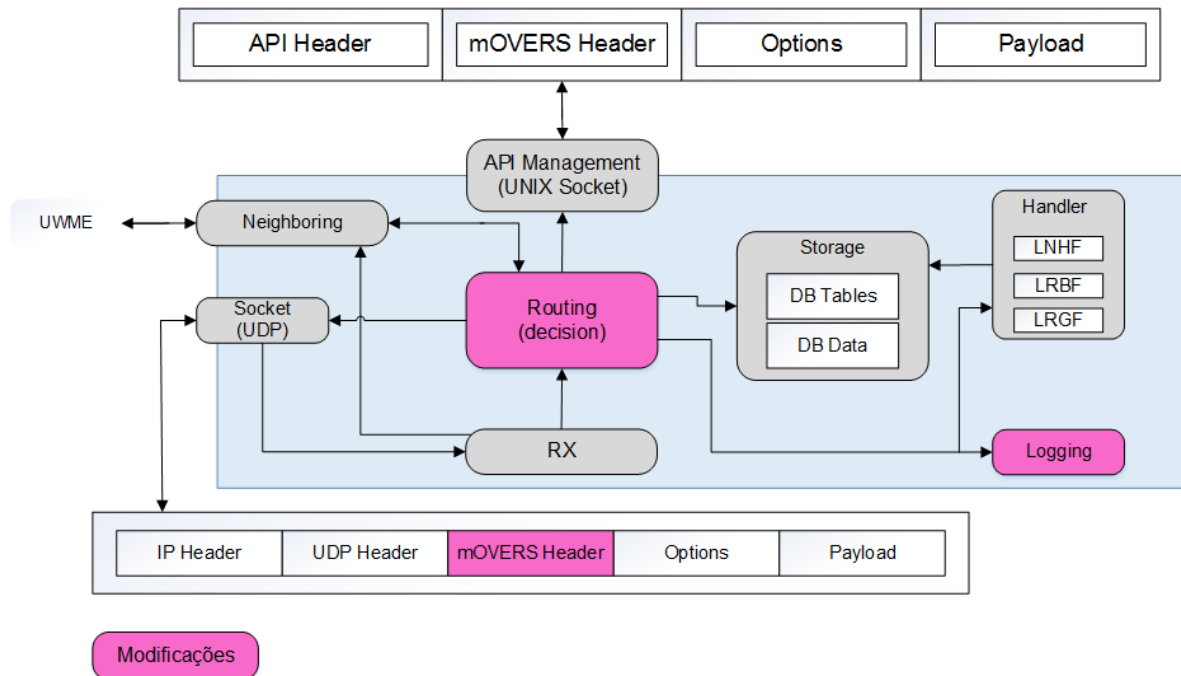


Figura 4.7: Arquitectura do mOVERS - modificações

Biblioteca de Suporte do Emulador DTN e Cabeçalho do Pacote

O cabeçalho do pacote, que se encontra na biblioteca de suporte, foi modificado por forma a integrar as duas estratégias de Alta e Baixa Densidade na DTN. As alterações foram as seguintes:

- **serviceID**: Este campo identifica o tipo de serviço do pacote. Foram acrescentados dois novos serviços: um para pacotes de emergência e outro para *Advertisement (ADV)*, ambos de emergência (Este campo foi modificado).
- **nNeigh**: O conceito inicial deste campo era conter o número de vizinhos que tinham recebido um determinado pacote. No caso de ser um anúncio de vizinhos, representava o número total de vizinhos que cada nó tinha. Com a introdução do conceito de pacotes de emergência, este campo, na estratégia de baixa densidade, serve para o nó que está a anunciar, informar os respetivos vizinhos (através dos pacotes de *advertisement*) do número de OBUs vizinhas que tem atrás em cada instante (este campo foi modificado).
- **nextMaxRankID**: Foi atribuído a cada nó na rede um *rank*, cujo significado varia nas duas estratégias (alta e baixa densidade). O nó que tiver o valor mais alto será o escolhido para encaminhar o pacote e tem a responsabilidade de calcular o seguinte. O primeiro nó que assume esta função será o que gera os pacotes e vai calcular a próxima OBU que vai disseminar os mesmos (este campo foi adicionado).
- **latitude**: Este campo contém a latitude do nó que gera os pacotes inicialmente. Esta informação serve para que todos os nós que receberam o pacote saibam a localização do acidente e para delimitar a zona de relevância, onde os pacotes vão ser disseminados (este campo foi adicionado).
- **longitude**: Este campo contém a longitude do nó que gera os pacotes inicialmente. Esta informação serve para que todos os nós que receberem o pacote saibam a localização do acidente e para delimitar a zona de relevância, onde os pacotes vão ser disseminados (este campo foi adicionado).
- **direção**: identifica a direcção do movimento de cada nó, em cada instante. É usado para filtrar os nós que se encontram atrás do acidente e dos respectivos *nextMaxRankID* (este campo foi adicionado).
- **latitude_vizinho**: Este campo contém a latitude de cada nó em cada instante (que é enviada nos pacotes de *advertisement*). Posteriormente, esta informação é guardada na estrutura de vizinhos. Esta informação serve para verificar se estes estão dentro da área de relevância e para determinar quem será o próximo nó a disseminar a mensagem (este campo foi adicionado).
- **longitude_vizinho**: Este campo contém a longitude de cada nó em cada instante (que é enviada nos pacotes de *advertisement*). Posteriormente, esta informação é guardada na estrutura de vizinhos. Esta informação serve para verificar se estes estão dentro da área de relevância e para determinar quem será o próximo nó a disseminar a mensagem (este campo foi adicionado).

- **direção_vizinho**: identifica a direção do movimento de cada nó, em cada instante (que é enviada nos pacotes de *advertisement*). Posteriormente, esta informação é guardada na estrutura de vizinhos. É usado para filtrar os nós vizinhos que se encontram na mesma direção, atrás do acidente e dos respectivos *nextMaxRankID* (este campo foi adicionado).
- **latitude_maxRank**: Este campo contém a latitude do nó escolhido para voltar a disseminar o pacote de emergência. Esta informação é uma referência que vai ser usada para os cálculos do campo *nextMaxRankID* (este campo foi adicionado).
- **longitude_maxRank**: Este campo contém a longitude do nó escolhido para voltar a disseminar o pacote de emergência. Esta informação é uma referência que vai ser usada para os cálculos do campo *nextMaxRankID* (este campo foi adicionado).

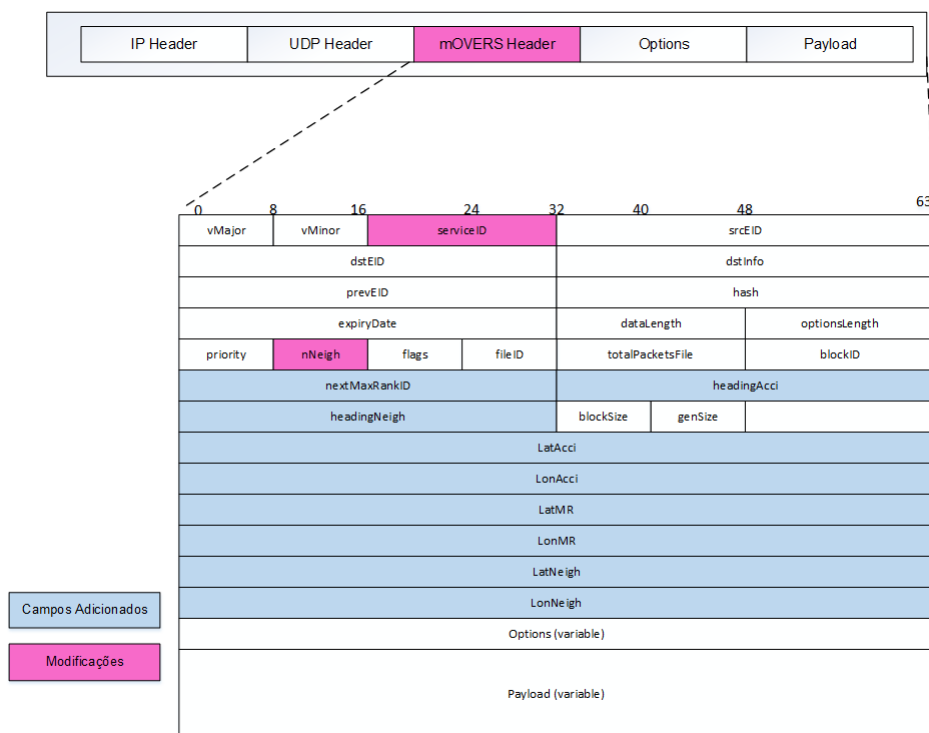


Figura 4.8: Estrutura do Pacote do mOVERS - modificações

Módulo Routing

Na classe *Routing* estão implementados os métodos que permitem a disseminação de mensagens de emergência através das três estratégias (como se pode ver na Figura 4.9). Nesta figura, observa-se a negrito os métodos implementados neste trabalho. A implementação no mOVERS das estratégias está dividida em duas lógicas: lógica de envio (**logic_v6()** corresponde à lógica de envio de pacotes de dados para a estratégia de alta densidade Figura 4.9, por exemplo) e lógica de receção (**pkt_v6(Packet*, DTNip, bool)** corresponde à lógica de receção de pacotes de dados para a estratégia de alta densidade Figura 4.9, por exemplo) de pacotes de emergência. É nestes métodos onde são tomadas todas as decisões relativamente

aos nós que enviam e quais nós recebem ou descartam os pacotes, bem como a delimitação da ZoR. Para além destes métodos, existem outras duas lógicas (envio e receção) para pacotes de *advertisement* (apenas para as estratégias de alta e baixa densidade), uma vez que estas duas estratégias necessitam de ter conhecimento da rede para os nós conseguirem tomar decisões. Nesta classe também foram implementados dois métodos que calculam o *MaxRank* de acordo com a estratégia pretendida, outro método que verifica se os nós estão dentro da ZoR e, por último, um método que indica quais os nós que estão atrás do acidente e sucessivamente do *MaxRank*.

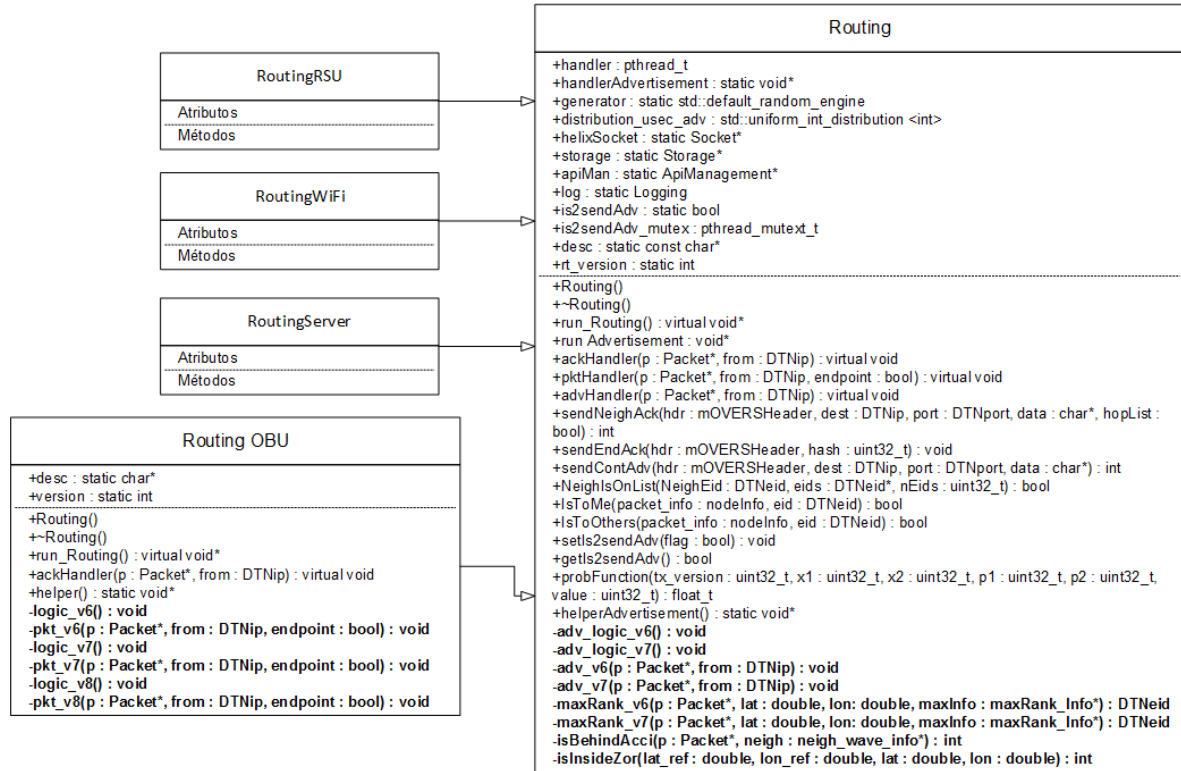


Figura 4.9: Classes de *Routing* do mOVERS

Módulo Storage

Existem vários métodos de *peek* já implementados neste módulo, na DTN com diversas finalidades tais como: devolver um pacote aleatório, por destinatário, pelo menor número de saltos e pelo número único que o identifica. O método de *peek* permite ir buscar um pacote à *storage* e transfere o mesmo da tabela *Expiry* para a tabela *onHold*, onde vai permanecer um período de tempo entre *ST_MIN_ONHOLD* e *ST_MAX_ONHOLD*, pré-definidos. No entanto, para facilitar a distinção entre pacotes de distribuição de conteúdos urgentes de conteúdos não urgentes, foi acrescentado um método novo (*peek_serviceID(serviceID)*) que filtra os pacotes pelo tipo de serviço. Desta forma, os nós ao receberem um dado pacote conseguem identificar o tipo de serviço e processá-lo de acordo com a sua prioridade.

Este método apenas vai buscar os pacotes que se encontram na tabela *Expiry*, isto é, uma lista que contém os pacotes ordenados por ordem crescente do "tempo de vida" dos mesmos. Contudo, este método não evita que os nós vão à tabela *OnHold* buscar pacotes que estejam em espera, repetidamente. Para resolver este problema, foi atribuído um valor elevado às variáveis de tempo mínimo e máximo de espera. Assim, durante esse intervalo de tempo, os pacotes não voltam a ser enviados.

- *peek_serviceID(uint16_t serviceID)*: Esta função devolve uma cópia do pacote guardado na *storage* pelo tipo de serviço do pacote.

Módulo Logging

O módulo *Logging* do mOVERS foi alterado para recolher dados que irão servir para avaliar o desempenho das estratégias de distribuição de conteúdos não urgentes. A classe relativa a este módulo está representada na Figura 4.10.

Logging
+handler : pthread_t
+log_path : static char
+full_path : static char
+desc : static const char
+logging_mutex : pthread_mutex_t
-packets_sent_per_timestamp : uint32_t
-packets_stored_per_timestamp : uint32_t
-packets_listen_per_timestamp : uint32_t
-control_packets_sent_per_timestamp : uint32_t
-packets_discard_per_timestamp : uint32_t
-packets_repeated_per_timestamp : uint32_t
-log_id : uint32_t
-timestamp : time_t
-last_timestamp : time_t
-generator : static std::default_random_engine
-distribution_usec : std::uniform_int_distribution <int>

Figura 4.10: Módulo de *Logging* alterado

O funcionamento do módulo de **Logging** está descrito na Figura 4.11. Como se vê na figura, o fluxograma pode seguir dois caminhos distintos: caso o modo Emulador esteja ligado, e caso o modo Emulador esteja desligado. Considerando a situação em que não está a ser utilizado o Emulador (isto é, a DTN está a correr nas placas usadas na rede real), o módulo de *logging* recolhe o tempo atual em que a execução da estratégia de disseminação está a ser feita. Caso contrário (o Emulador está ligado), o *timestamp* recolhido corresponde ao *timestamp* presente nos dados de entrada do emulador (contido nos dados que estão na base de dados fornecida ao emulador para a execução das estratégias). Sempre que o *timestamp* é diferente do anterior, este é atualizado para o *timestamp* seguinte e os dados correspondentes a esse *timestamp* são guardados num ficheiro. Estes dados serão, posteriormente, utilizados para analisar o comportamento das estratégias de disseminação de mensagens de emergência. No fim, a *thread* espera um tempo aleatório entre MIN_TIME_STEP_LOG e MAX_TIME_STEP_LOG. Este tempo aleatório serve para que os processos consigam aceder paralelamente e de forma concorrente ao módulo.

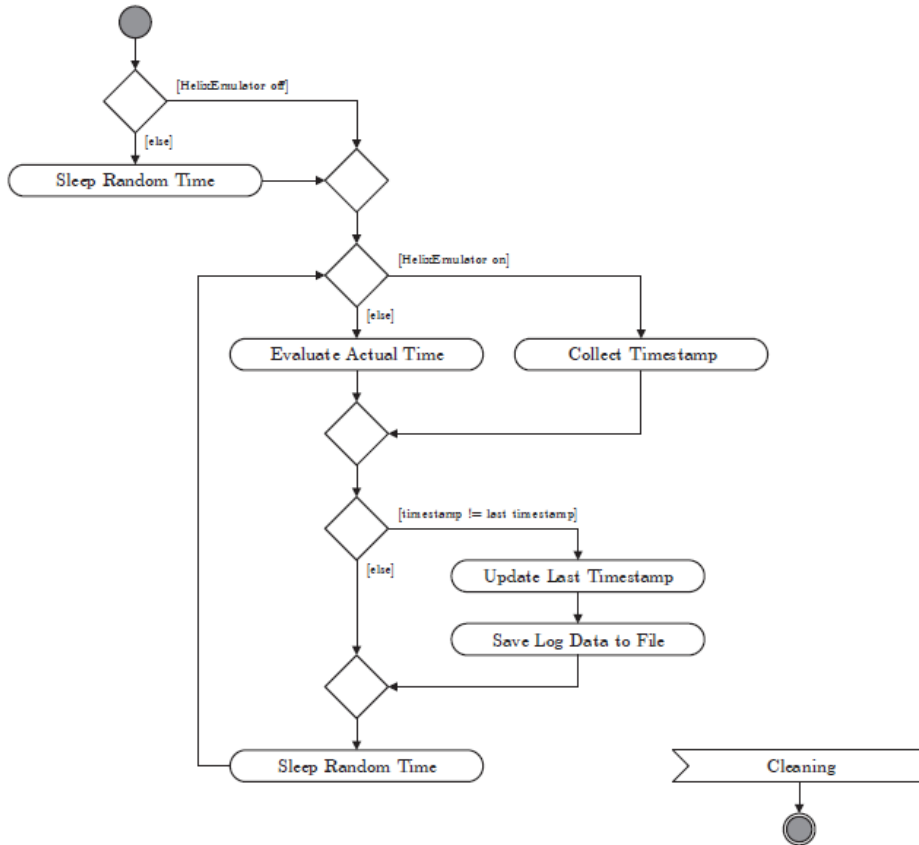


Figura 4.11: Fluxograma do Funcionamento do Módulo de *Logging* [27]

Alguns destes dados (*node_EID*, *timestamp*, *packets_sent_per_timestamp*, *packets_listen_per_timestamp* e *control_packets_sent_per_timestamp*) já existiam no mOVERS, mas para as estratégias de disseminação de conteúdos não urgentes. Portanto foi necessário alterá-los para as estratégias de disseminação de mensagens de emergência. Os dados que

foram acrescentados (`packets_sent_per_timestamp`, `packets_discard_per_timestamp` e `packets_repeated_per_timestamp`) têm como finalidade perceber a forma como é feito o encaminhamento dos pacotes de emergência e avaliar o desempenho das estratégias de disseminação de mensagens de emergência.

Os dados que foram tirados ao longo do processo de disseminação de mensagens de emergência são apresentados de seguida:

- `log_id`: identifica o número único da entrada no ficheiro *log*.
- `node_EID`: identifica o ID de cada nó da rede que gera a informação.
- `timestamp`: representa o instante temporal em que cada iteração foi executada.
- `packets_sent_per_timestamp`: indica o número total de pacotes enviados por cada nó, na ZoR, em cada instante temporal do processo de disseminação.
- `packets_stored_per_timestamp`: indica o número total de pacotes úteis recebidos por cada nó que está na ZoR, em cada instante temporal do processo de disseminação.
- `packets_listen_per_timestamp`: indica o número total de pacotes "ouvidos" por cada nó que está na ZoR, em cada instante temporal do processo de disseminação.
- `control_packets_sent_per_timestamp`: indica o número total de pacotes de *advertisement* enviados por cada nó, em cada instante temporal do processo de disseminação.
- `packets_discard_per_timestamp`: indica o número total de pacotes descartados (são recebidos, mas não são guardados) por cada nó que está fora da ZoR, em cada instante temporal do processo de disseminação.
- `packets_repeated_per_timestamp`: indica o número total de pacotes repetidos, em cada instante temporal do processo de disseminação.

4.3.2 Restrições do mOVERS para a disseminação de mensagens de emergência

O mOVERS não estava preparado para a disseminação de mensagens de emergência e, portanto alguns módulos referidos anteriormente (*Routing* e *Logging*, por exemplo) tiveram de ser modificados. Existiam também lacunas no que diz respeito à inexistência de algumas estruturas que permitissem ter acesso às informações essenciais de cada nó (latitude, longitude e direção do movimento), de forma a ser possível calcular os *MaxRanks*, de acordo com a estratégia. Quanto às estratégias de encaminhamento foram implementadas e integradas três, pois as existentes no mOVERS não satisfaziam as condições necessárias para a disseminação de mensagens de emergência. De forma a ser possível introduzir o novo conceito de *MaxRank* com o critério de seleção baseado na escolha do nó mais distante do nó fonte e sucessivamente dos *MaxRanks*, foi necessário alterar os pacotes de *advertisement*, para que estes enviassem a posição de cada nó. Assim, conhecendo as posições em cada instante de cada nó, é possível calcular as distâncias a que cada vizinho se encontra do nó fonte, por exemplo, e determinar qual o primeiro *MaxRank*. No entanto, para calcular os *MaxRanks* seguintes, surgiu a necessidade de criar uma estrutura que guardasse as informações (latitude, longitude, heading e ID)

do *MaxRank* para depois comparar com as informações dos nós vizinhos. Por sua vez, para o critério de seleção da estratégia de Baixa Densidade, para saber quantos vizinhos cada nó tem atrás, foi criada uma nova estrutura com o objetivo de guardar e atualizar essa informação, entre outras. Desta forma, reuniram-se todas as condições para calcular o *MaxRank* nas duas estratégias, determinar quais os nós que estavam dentro e fora da ZoR e quais destes se encontravam na mesma direção do movimento do acidente, pois já era possível conhecer todas as informações necessárias sobre todos os nós. Quanto às três estratégias de disseminação de mensagens de emergência, foram implementadas duas lógicas (envio e recepção de pacotes de emergência) para cada estratégia. Mais uma vez, o cabeçalho do pacote não estava adequado para situações de emergência e foi modificado de acordo com a Figura 4.8. Com a existência de dois tipos de disseminação de conteúdos no mOVERS (urgente e não urgente), foram adicionados dois tipos de serviços de emergência, um para pacotes de dados e outro para pacotes de *advertisement*.

4.4 Considerações do Capítulo

O mOVERS já estava preparado para lidar com distribuição de conteúdos utilizando vários processos em paralelo. De forma a ser possível integrar disseminação de conteúdos urgentes, foram implementadas três novas estratégias, muito diferentes das existentes para distribuição de conteúdos. Foi também necessário criar novas estruturas para lidar com o novo conceito de *MaxRank*, bem como para conseguir extrair mais informação acerca dos nós, como por exemplo, as coordenadas GPS. Por fim, foram acrescentadas algumas variáveis e métodos correspondentes ao módulo já existente, *Logging*, para ser possível recolher informação durante o processo de disseminação de conteúdos urgentes.

Capítulo 5

Testes e Resultados

5.1 Descrição do Capítulo

No capítulo anterior foram descritos os módulos que constituem o mOVERS, as suas funcionalidades e a arquitetura do emulador utilizado para a implementação das três estratégias de disseminação de mensagens de emergência. Foi também explicada a forma como é feita a recolha de dados da base de dados existente e como os dados estão guardados na mesma, utilizando três tabelas auxiliares. Além disso, foram descritas algumas restrições existentes no mOVERS, no que diz respeito ao encaminhamento de mensagens de emergência, assim como as modificações feitas nos módulos já existentes do emulador para a implementação das três estratégias de disseminação de mensagens de emergência.

Neste capítulo será apresentado e descrito o *Software* e equipamento usado para correr as várias emulações, correspondentes aos cenários aplicados às três estratégias desenvolvidas. Serão explicados os vários cenários estáticos usados para obter os resultados apresentados e a justificação para a escolha dos valores dos comprimentos das ZoR e do alcance WAVE. De seguida serão apresentados os resultados obtidos para os 6 cenários (gerados aleatoriamente) estudados, e será feita uma comparação acerca do comportamento das estratégias, em relação a cada cenário e às mudanças efetuadas entre cenários. Por fim, conclui-se qual a estratégia que apresenta melhores resultados, tendo em conta as métricas utilizadas.

5.2 *Software* e Equipamento

As três estratégias de disseminação de mensagens de emergência foram desenvolvidas num computador pessoal. No entanto, as experiências e testes realizados no mOVERS foram feitos numa máquina virtual, uma vez que o emulador (mOVERS) se encontra num servidor contido numa máquina virtual. As especificações da máquina virtual utilizada para realizar os testes correspondentes às três estratégias estão descritas de seguida:

- Sistema Operativo: Linux 2.6/3.x/4.x (64-bit)
- Processador: Intel[®] Core[™] i5-5200U Central Processing Unit (CPU) @ 2.20 GHz
- RAM: 6 GB

5.3 Scripts

Os resultados obtidos para a avaliação do desempenho das três estratégias estão representados em gráficos elaborados através de *scripts* feitos em *Matlab* [50]. Os dados utilizados para obter os gráficos são as informações recolhidas pelo módulo de *Logging* (Secção 4.2.3 e Figura 4.10) ao longo dos processos de emulação para as várias ZoRs e para os vários veículos (OBUs) associados a cada uma das três estratégias.

5.4 Descrição do Cenário

Os cenários utilizados ao longo de todas as experiências foram concebidos da seguinte forma: para cada ZoR foram escolhidas cinco quantidades diferentes de veículos (por exemplo, para a ZoR de 3Km existem 5 pontos que correspondem a 15, 20, 30, 40 e 50 veículos), de forma a ir aumentando a densidade dos cenários e ser possível observar o comportamento e o desempenho das três estratégias de disseminação de mensagens de emergência desenvolvidas. Vão ser considerados 6 cenários: ZoR de 3Km e alcance WAVE de 500m; ZoR de 3Km e alcance WAVE de 700m; ZoR de 5Km e alcance WAVE de 500m; ZoR de 5Km e alcance WAVE de 700m; ZoR de 7Km e alcance WAVE de 500m; ZoR 7Km de e alcance WAVE de 700m. Cada um corresponde ao conjunto de 3 gráficos: taxa de entrega de pacotes ($Delivery Ratio = \frac{N^o \text{ de pacotes recebidos}}{N^o \text{ de pacotes enviados}} \times 100$), taxa de pacotes redundantes ($Overhead = \frac{N^o \text{ de pacotes repetidos}}{N^o \text{ de pacotes ouvidos}} \times 100$) e número de iterações que corresponde ao número de saltos pelos quais o pacote passou. Para cada cenário *versus* quantidade de veículos foram gerados 10 conjuntos de informações (coordenadas, quantidade de vizinhos, entre outros). Isto permite avaliar diferentes "fotografias", correspondentes aos 10 conjuntos de dados aleatórios e, desta forma, verificar o comportamento das estratégias para cada cenário, mantendo a quantidade de veículos. Os tamanhos das três ZoRs foram escolhidos tendo em conta os valores mais usados e encontrados na literatura (2, 4 e 6 Km) [44], [51], [52]. No entanto, foram escolhidos valores superiores (mais 1 Km), uma vez que os valores do alcance do WAVE usados nesta dissertação também são superiores aos valores mais comuns na literatura [44], [51], [52]. Nos cenários considerados as OBUs que estiverem fora da ZoR irão descartar todos os pacotes recebidos. Portanto é de salientar que os valores que se encontram nos eixos dos gráficos da Secção 5.5 correspondem às OBUs que foram utilizadas em cada teste (dentro e fora da ZoR). No entanto, as percentagens obtidas são referentes apenas aquelas que estão dentro da ZoR. Portanto, as percentagens correspondentes à taxa de entrega (*Delivery Ratio*) e taxa de pacotes repetidos (*Overhead*) são a média dos 10 conjuntos de dados para cada quantidade de veículos (OBUs). Nestes cenários considera-se que os veículos se encontram parados no instante temporal em que se dá o acidente, isto é, em cada cenário é analisado o comportamento das três estratégias, tendo em conta as diferentes "fotografia" do instante do acidente, tendo em conta que o mesmo ocorreu num cenário de auto estrada.

Nos cenários considerados foi feita uma avaliação das três estratégias de disseminação de mensagens de emergência, tendo em conta o aumento da quantidade de veículos nas três ZoRs. Foi assumido que ocorreu uma situação de emergência, como por exemplo um acidente e, a partir daí, foram testadas as três estratégias de disseminação. Como não foi feito um estudo exaustivo relativamente à densidade de veículos em cada cenário, não foi possível definir os limites de alta e baixa densidade, tendo sido apenas estudado o comportamento das três estratégias com o aumento progressivo da densidade.

5.5 Avaliação do Desempenho das Estratégias

Os cenários avaliados de seguida dizem respeito a 3 ZoRs e, para cada uma, o alcance WAVE assume os valores 500 e 700 metros. A métrica referente às iterações utilizadas por cada estratégia poderá indicar o tempo de disseminação de mensagens de emergência, isto é, quantas mais iterações existirem, maior será o tempo de disseminação. Os resultados obtidos são apresentados com intervalos de confiança de 90%.

5.5.1 Zona de Relevância de 3Km

Neste cenário é considerada uma ZoR com 3 Km e WAVE com alcance de 500 metros. Através da Figura 5.1 é possível observar que a quantidade de iterações necessárias para entregar o pacote aumenta, à medida que o número de veículos considerados na ZoR também aumenta (estratégias de Alta Densidade e *Broadcast*). Esta tendência verifica-se apenas para as estratégias de Alta Densidade e *Broadcast* e a quantidade de iterações necessárias está relacionada com a taxa de entrega de pacotes dessas mesmas estratégias. Quando a taxa de entrega de pacotes é semelhante (20 e 30 veículos, por exemplo), a média do número de iterações varia pouco, isto é, mantém praticamente constante. Quando a taxa de entrega aumenta significativamente (15 e 20 veículos), a média das iterações aumenta bruscamente também. Relativamente à estratégia de Baixa Densidade, a média de iterações varia de 2.5 a 4, uma vez que esta estratégia apresenta uma taxa de entrega de pacotes de emergência baixa, comparando com as outras duas estratégias de disseminação de mensagens de emergência. Novamente, é possível relacionar a tendência decrescente da média de iterações com a taxa de entrega. Quanto à estratégia de Baixa Densidade, observa-se na Figura 5.1 que a quantidade de iterações é muito baixa. A percentagem de pacotes entregues também é baixa face às outras duas estratégias, pois o *MaxRank* selecionado pelo *MaxRank* da última iteração não consegue assumir-se como nó retransmissor, isto é, vai considerar o pacote como repetido uma vez que já o recebeu anteriormente. Devido a este facto, a estratégia de Baixa Densidade torna-se a menos ineficiente das três estratégias propostas, pois se o nó escolhido para retransmitir já tiver recebido esse pacote de outro *MaxRank*, vai considerá-lo repetido e, por isso, não consegue assumir-se como *MaxRank*. Portanto, como o critério de seleção desta estratégia é feito consoante o número de vizinhos que cada nó tem atrás, quanto maior for a quantidade de veículos atrás, maior será a possibilidade de o nó retransmissor escolhido já ter recebido o pacote anteriormente.

Portanto, pode ver-se na Figura 5.2 que, as curvas correspondentes às estratégias de Alta Densidade e *Broadcast* têm uma tendência de crescimento que se deve ao aumento do número de veículos na ZoR, pois quanto maior for o número de veículos, menor será a distância entre eles e menor será a probabilidade de existir um ou mais veículos isolados.

Observando a Figura 5.3, verifica-se que a tendência da curva correspondente à estratégia de *Broadcast* é de crescimento, à medida que a quantidade de veículos aumenta. Esta estratégia apresenta percentagens bastante elevadas, pois todos os veículos contidos na ZoR enviam um pacote de emergência para todos os seus vizinhos. Em relação à curva correspondente à estratégia de Alta Densidade, a taxa de pacotes repetidos na rede mantém-se praticamente constante e perto dos 50%, pois o critério de seleção do *MaxRank* é feito escolhendo o (veículo) vizinho mais distante, dentro do alcance do WAVE. Por fim, a taxa de pacotes repetidos da curva que corresponde à estratégia de Baixa Densidade não segue uma tendência, pois o critério de seleção desta estratégia depende da quantidade de vizinhos que cada veículo tem

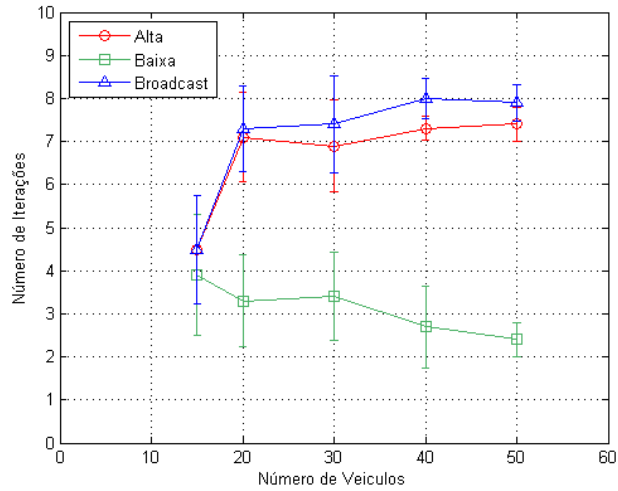


Figura 5.1: Avaliação do desempenho para a ZoR de 3 Km e WAVE de 500m: número de iterações

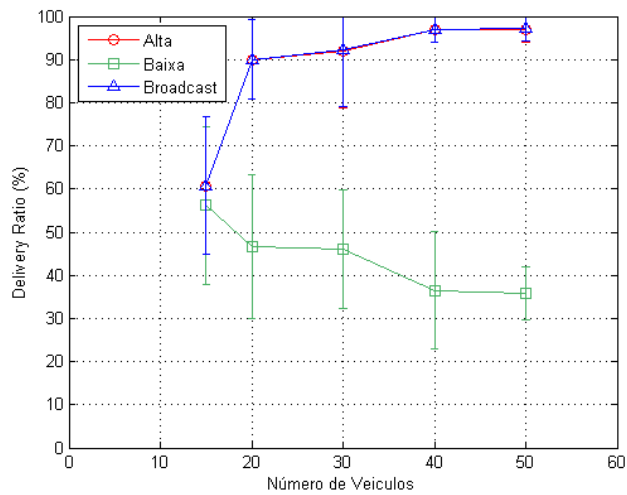


Figura 5.2: Avaliação do desempenho para a ZoR de 3 Km e WAVE de 500m: taxa de entrega

atrás de si. É também visível que a percentagem de pacotes repetidos é elevada, uma vez que a taxa de pacotes entregues não ultrapassa os 40%.

No cenário seguinte é considerada uma ZoR com 3 Km e WAVE com alcance de 700 metros. Quando o alcance do WAVE aumenta de 500 para 700 metros, a tendência das curvas das iterações correspondentes às estratégias de Alta Densidade e *Broadcast* cresce mais lentamente. No entanto, verifica-se novamente que existe uma relação entre as Figuras 5.5 e 5.4, pois quanto maior for a taxa de entrega de pacotes, mais iterações serão necessárias para entregar o pacote de emergência. Quanto à estratégia de Baixa Densidade, é possível ver que a média de iterações necessárias é muito baixa e varia muito pouco (entre 2 e 3 iterações), uma vez que a taxa de entrega ao longo dos 5 pontos da ZoR é sempre baixa.

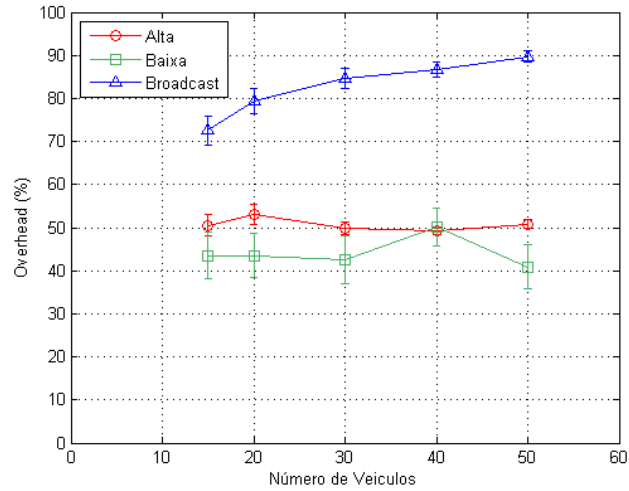


Figura 5.3: Avaliação do desempenho para a ZoR de 3 Km e WAVE de 500m: taxa de pacotes repetidos

Mais uma vez observa-se que a taxa de entrega de pacotes aumenta, quanto mais elevada for a quantidade de veículos na ZoR, para as estratégias de Alta Densidade e *Broadcast*.

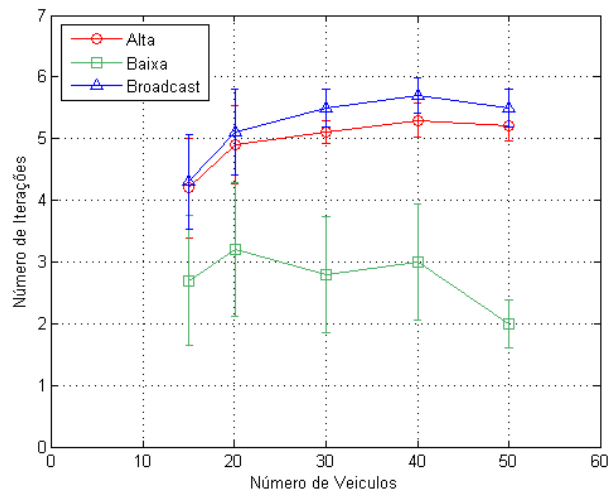


Figura 5.4: Avaliação do desempenho para a ZoR de 3 Km e WAVE de 700m: número de iterações

Em relação à taxa de pacotes repetidos na rede, é possível ver na Figura 5.6 que a curva da estratégia de *Broadcast* segue, novamente uma tendência de crescimento, embora menos acentuada. Quanto à curva da estratégia de Alta Densidade, a taxa de pacotes repetidos mantém-se praticamente constante, pelas razões referidas anteriormente para um alcance WAVE de 500 metros. Por fim, observa-se que a taxa de pacotes repetidos correspondente à estratégia de Baixa Densidade é elevada, tendo em conta a baixa taxa de entrega de pacotes.

Portanto, a melhor estratégia, tanto para um alcance WAVE de 500 metros como para um

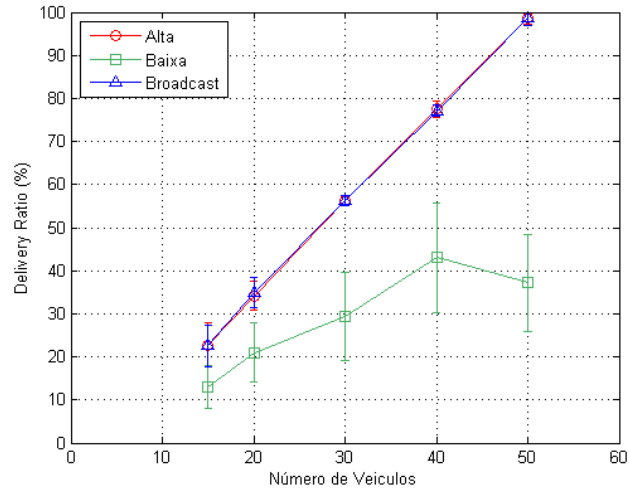


Figura 5.5: Avaliação do desempenho para a ZoR de 3 Km e WAVE de 700m: taxa de entrega

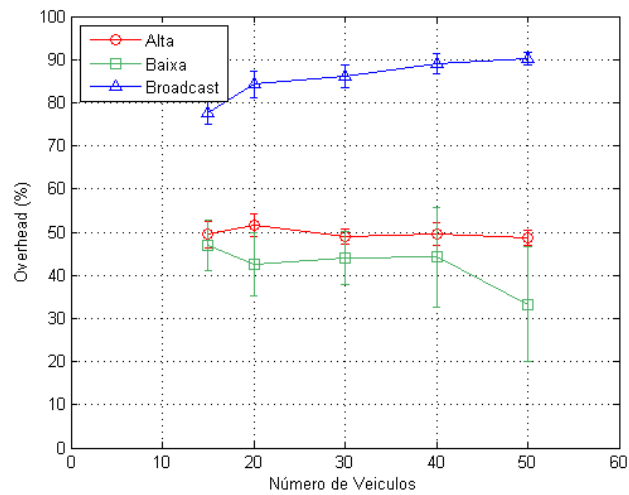


Figura 5.6: Avaliação do desempenho para a ZoR de 3 Km e WAVE de 700m: taxa de pacotes repetidos

alcance de 700 metros, é a de Alta Densidade, pois apesar de apresentar uma taxa elevada de pacotes de emergência (tal como a estratégia de *Broadcast*), a taxa de pacotes repetidos da estratégia de Alta Densidade é inferior (cerca de 30 a 40%) à da estratégia de *Broadcast*.

5.5.2 Zona de Relevância de 5Km

Neste cenário é considerada uma ZoR com 5 Km e WAVE com alcance de 500 metros. Observando a Figura 5.7 vê-se que a média de iterações necessárias para entregar o pacote de emergência é mais elevada na estratégia de *Broadcast* do que na de Alta Densidade. Isto deve-se ao facto de na estratégia de *Broadcast*, poder existir na mesma iteração mais do que um veículo a enviar o pacote e de todos os veículos enviarem pacotes para todos os seus

vizinhos. Estas situações não se verificam na estratégia de Alta Densidade, pois existe um critério de seleção que restringe o número de veículos que podem enviar pacotes de emergência. Esta tendência verifica-se praticamente em todos os pontos de todas as ZoRs consideradas, independentemente do alcance WAVE considerado (500 ou 700 metros).

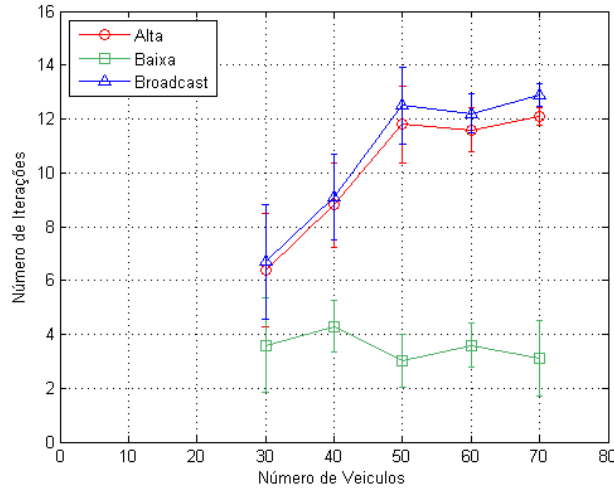


Figura 5.7: Avaliação do desempenho para a ZoR de 5 Km e WAVE de 500m: número de iterações

Outra tendência que se observa, relativamente à estratégia de Baixa Densidade, é a baixa média das iterações necessárias, pois a taxa de entrega também é sempre muito baixa, comparando com as outras duas estratégias. Quanto à tendência das curvas da taxa de entrega correspondentes às estratégias de Alta Densidade e *Broadcast*, observa-se que é de crescimento ao longo da Figura 5.8.

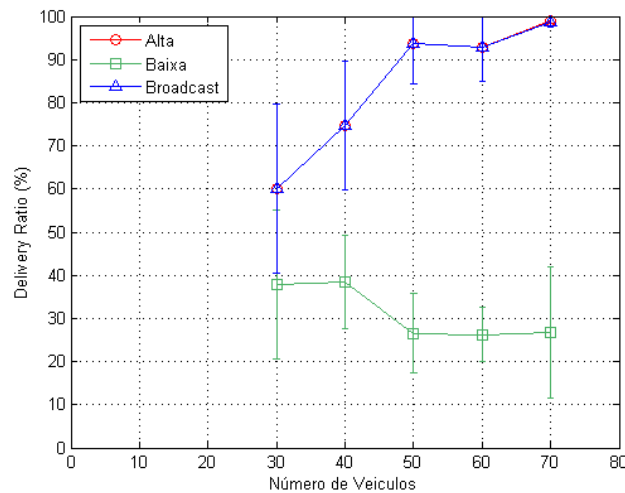


Figura 5.8: Avaliação do desempenho para a ZoR de 5 Km e WAVE de 500m: taxa de entrega

Através da Figura 5.9 verifica-se que a taxa de pacotes repetidos na rede, considerando a

estratégia de *Broadcast*, aumenta com o número de veículos contidos na ZoR. Isto significa que, quanto maior for a quantidade de veículos contidos na ZoR, maior será a taxa de pacotes repetidos, pois todos esses veículos enviam pacotes de emergência para todos os seus vizinhos. Ao observar a curva da estratégia de Alta Densidade, vê-se que a percentagem de pacotes repetidos aproxima-se de 50%, havendo porém um valor próximo de 55%. Quanto à curva da estratégia de Baixa Densidade, pode afirmar-se que a percentagem de pacotes repetidos é elevada, tendo em conta a baixa percentagem de pacotes entregues.

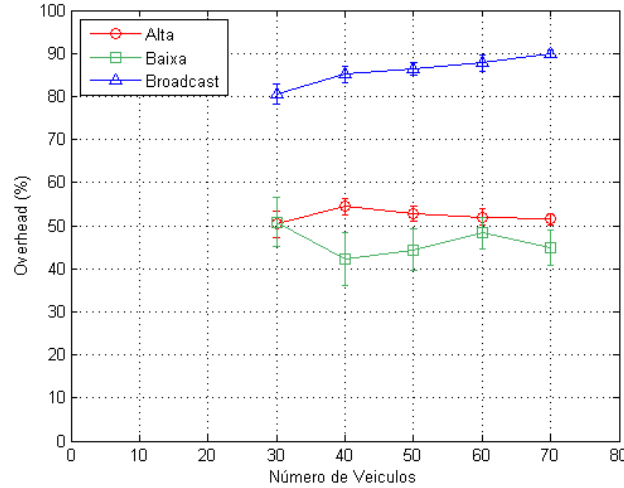


Figura 5.9: Avaliação do desempenho para a ZoR de 5 Km e WAVE de 500m: taxa de pacotes repetidos

No cenário seguinte é considerada uma ZoR com 5 Km e WAVE com alcance de 700 metros. Quando o alcance WAVE aumenta para 700 metros, verifica-se que são necessárias menos iterações, comparando a Figura 5.7, onde o alcance WAVE é de 500 metros. O facto de existir uma quebra no aumento das iterações entre os pontos 50 e 60 (Alta Densidade), pode dever-se à existência de mais veículos fora da ZoR quando foram gerados os 10 conjuntos de dados para 60 veículos. Portanto existirão menos veículos dentro da ZoR, fazendo com que sejam necessárias menos iterações quando foram considerados 60 veículos nas experiências. A tendência da curva da estratégia de *Broadcast* é de crescimento, embora seja pouco acentuado. A média de iterações da estratégia de Baixa Densidade varia apenas entre 3 e 4, pois a taxa de entrega de pacotes varia aproximadamente entre 30 e 40% apenas, ou seja, é baixa.

Quanto à taxa de entrega de pacotes, verifica-se que, para ambas as estratégias (Alta Densidade e Broadcast), a tendência das curvas é de crescimento, aproximando-se de 100%, nos pontos de 60 e 70 veículos. Contrariamente, a percentagem de pacotes entregues na estratégia de Baixa Densidade é bastante inferior em relação às outras duas estratégias, como já foi referido anteriormente.

Em relação à Figura 5.12 é possível ver-se que a tendência da curva da estratégia de *Broadcast* é de crescimento e a percentagem de pacotes repetidos é bastante elevada, como se verifica ao longo de todos pontos de todas as ZoRs. Mais uma vez, a percentagem de pacotes repetidos na estratégia de Alta Densidade aproxima-se dos 50%, exceto no ponto 30, onde se aproxima de 60%. Relativamente à estratégia de Baixa Densidade pode afirmar-se que a percentagem de pacotes repetidos na rede é novamente elevada, tendo em conta que a taxa

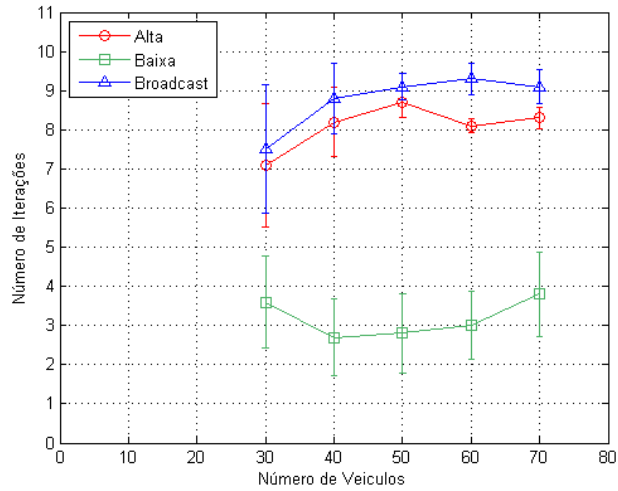


Figura 5.10: Avaliação do desempenho para a ZoR de 5 Km e WAVE de 700m: número de iterações

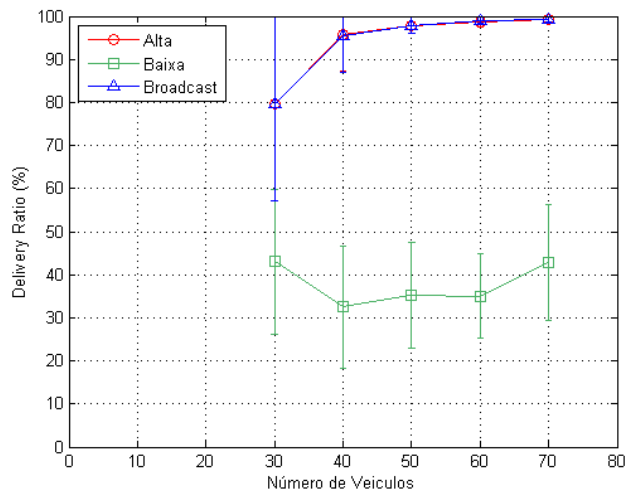


Figura 5.11: Avaliação do desempenho para a ZoR de 5 Km e WAVE de 700m: taxa de entrega

de entrega de pacotes da mesma é baixa.

Portanto, a melhor estratégia, tanto para um alcance WAVE de 500 metros como para um alcance de 700 metros, é a de Alta Densidade, pois apesar de apresentar uma taxa elevada de pacotes de emergência (tal como a estratégia de *Broadcast*), a taxa de pacotes repetidos da estratégia de Alta Densidade é inferior (cerca de 30 a 40%) à da estratégia de *Broadcast*.

5.5.3 Zona de Relevância de 7Km

Neste cenário é considerada uma ZoR com 7 Km e WAVE com alcance de 500 metros. Através da Figura 5.13 é possível observar-se que a quantidade de iterações necessárias, con-

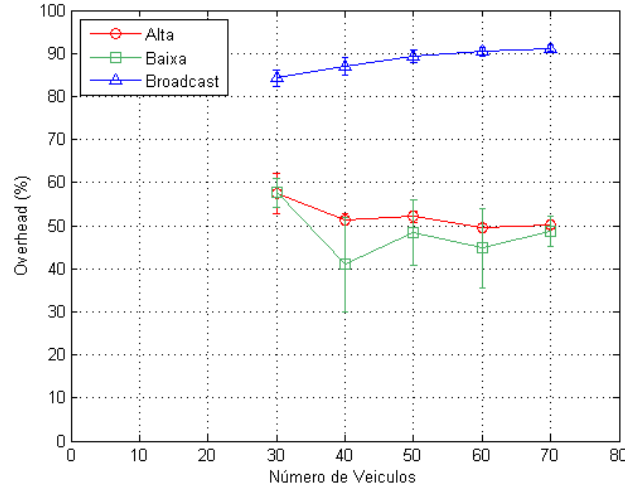


Figura 5.12: Avaliação do desempenho para a ZoR de 5 Km e WAVE de 700m: taxa de pacotes repetidos

siderando as estratégias de Alta Densidade e *Broadcast* aumenta, à medida que o número de veículos contidos na ZoR também aumenta. O ligeiro decréscimo observado entre os pontos 70 e 80 (veículos) pode relacionar-se com a taxa de entrega constante nesses mesmos pontos (Figura 5.14). Como seria de esperar, a média das iterações correspondentes à estratégia de Baixa Densidade é baixa, uma vez que a taxa de entrega desta estratégia também é muito baixa, aproximando-se dos 20% apenas.

Quanto às curvas das estratégias de Alta Densidade e *Broadcast*, ambas seguem uma tendência de crescimento, exceto no último ponto, onde se mantém constante. Isto pode dever-se ao facto de, na estratégia de Alta Densidade, a escolha dos *MaxRanks* ter condicionado a entrega dos pacotes por todos os veículos. Isto significa que o *MaxRank* escolhido poderá não ter vizinhos atrás que ainda não tenham recebido o pacote, ou seja, existe uma quebra e, por consequência a execução da estratégia de Alta Densidade termina. É também por este motivo que o ponto 40 da estratégia de *Broadcast* tem uma taxa de entrega superior à da estratégia de Alta Densidade.

Analisando a Figura 5.15 verifica-se que a curva correspondente à estratégia de *Broadcast* tem uma tendência crescente e é bastante mais elevada do que a curva correspondente à estratégia de Alta Densidade, sendo que a percentagem de pacotes repetidos da estratégia de Alta Densidade varia entre 50% e 60% aproximadamente e a da estratégia de *Broadcast* varia entre 80% e 90%. Finalmente, em relação à estratégia de Baixa Densidade, observa-se que a percentagem de pacotes repetidos é bastante elevada, tendo em conta a taxa de entrega de pacotes bastante reduzida. O pico existente no ponto 60 veículos deve-se ao facto de quase todos os conjuntos de dados gerados aleatoriamente terem uma percentagem de pacotes repetidos superior a 60%. Uma vez que o método de seleção dos *MaxRanks* usa a escolha do nó que tiver mais vizinhos atrás, isto é, se os vizinhos do *MaxRank* forem os mesmos ou tiverem muitos em comum com o próximo nó a ser escolhido, a taxa de pacotes repetidos torna-se elevada. Este pico já não se verifica quando o alcance WAVE aumenta para 700m, pois os *MaxRanks* podem ter mais vizinhos e à partida diferentes, ou também pode acontecer haver uma mudança na escolha de *MaxRanks* e, portanto esse resultado altera-se.

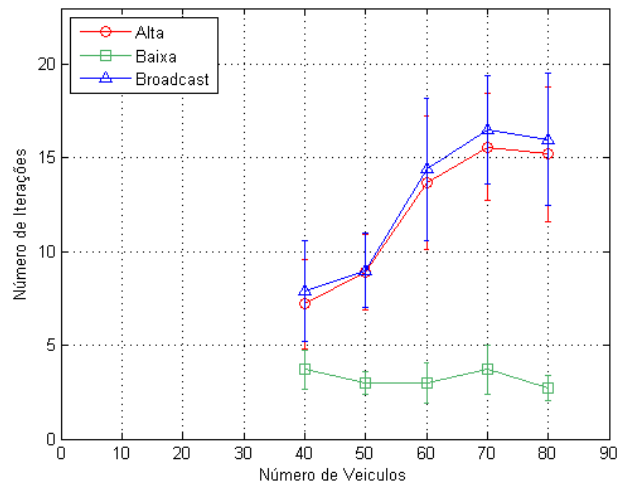


Figura 5.13: Avaliação do desempenho para a ZoR de 7 Km e WAVE de 500m: número de iterações

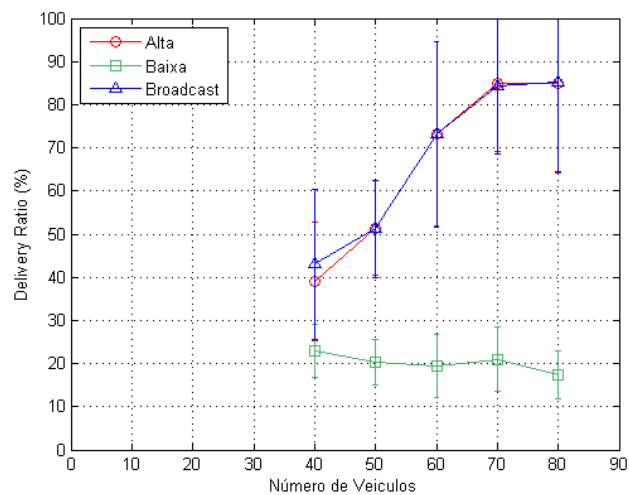


Figura 5.14: Avaliação do desempenho para a ZoR de 7 Km e WAVE de 500m: taxa de entrega

No cenário seguinte é considerada uma ZoR com 7 Km e WAVE com alcance de 700 metros. Analisando a Figura 5.16, verifica-se que a estratégia de *Broadcast* necessita, em média, de mais iterações do que a estratégia de Alta Densidade. As duas curvas correspondentes a essas duas estratégias têm uma tendência de crescimento e pode ver-se que, para um alcance WAVE maior (700 metros), a quantidade de iterações é menor. Quanto à estratégia de Baixa Densidade, o número de iterações praticamente não varia e é muito baixo, uma vez que a taxa de entrega de pacotes de emergência desta estratégia também é muito baixa.

Relativamente à Figura 5.17, é possível ver que as curvas sobrepostas das estratégias Alta Densidade e *Broadcast* têm uma tendência de crescimento, atingindo praticamente os 100%

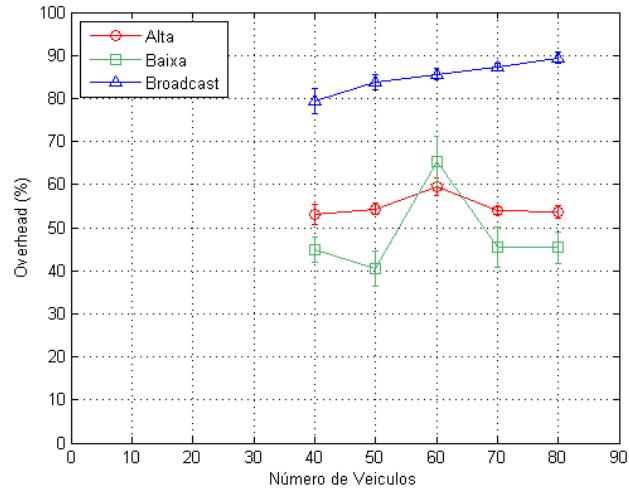


Figura 5.15: Avaliação do desempenho para a ZoR de 7 Km e WAVE de 500m: taxa de pacotes repetidos

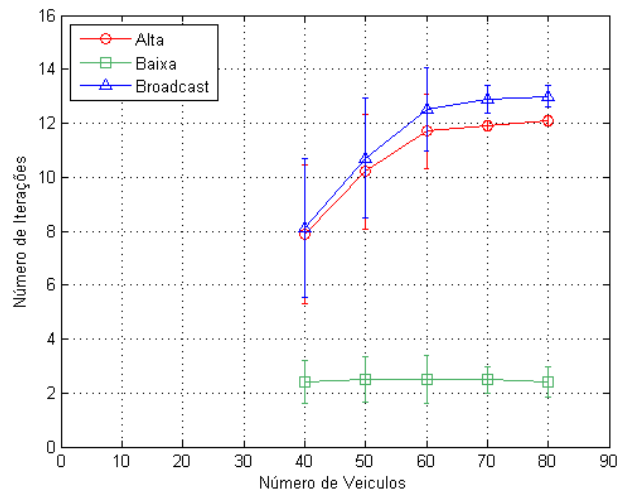


Figura 5.16: Avaliação do desempenho para a ZoR de 7 Km e WAVE de 700m: número de iterações

nos pontos 70 e 80 veículos. Comparando com a Figura 5.13 é visível que a percentagem de pacotes entregues da Figura 5.16 é mais elevada em todos os pontos.

Por fim, a Figura 5.18 mostra que a estratégia de *Broadcast* tem uma percentagem de pacotes repetidos muito superior à estratégia de Alta Densidade. Esta situação verifica-se em todos os cenários de todas as ZoRs, pois na estratégia de *Broadcast* todos os veículos dentro da ZoR enviam pacotes de emergência para todos os seus vizinhos. Quanto à estratégia de Alta Densidade, esta percentagem é praticamente constante e próxima de 50%, como acontece em quase todos os cenários. É ainda possível verificar que a estratégia de Baixa Densidade possui maior percentagem de pacotes repetidos (cerca de 40%) do que de pacotes entregues

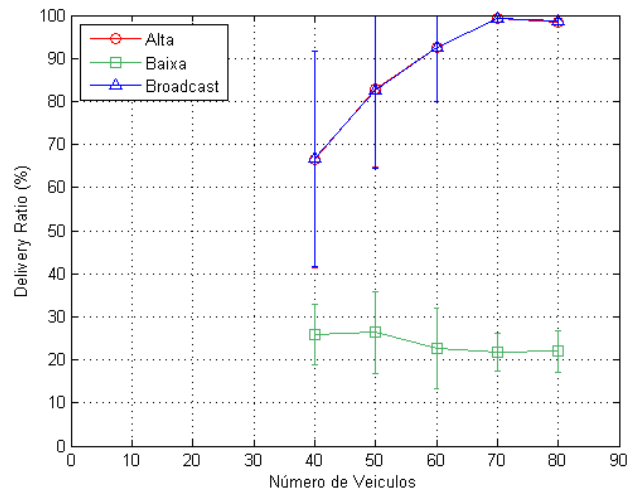


Figura 5.17: Avaliação do desempenho para a ZoR de 7 Km e WAVE de 700m: taxa de entrega

(cerca de 25%), situação que se verifica para todas as situações, exceto na Figura 5.3.

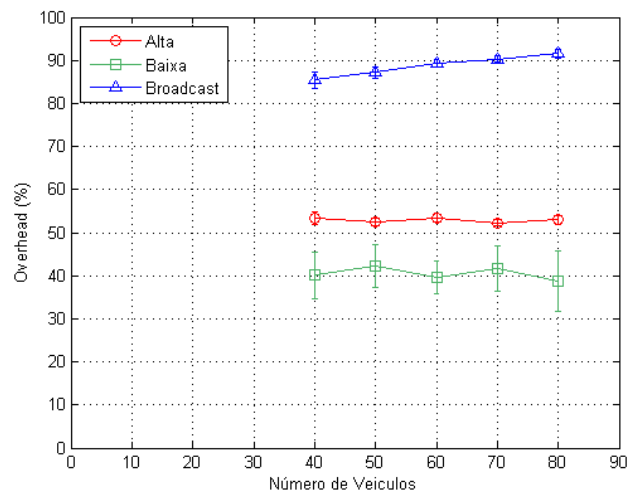


Figura 5.18: Avaliação do desempenho para a ZoR de 7 Km e WAVE de 700m: taxa de pacotes repetidos

Portanto, a melhor estratégia, tanto para um alcance WAVE de 500 metros como para um alcance de 700 metros, é a de Alta Densidade, uma vez que apresenta uma taxa elevada de pacotes de emergência e uma taxa de pacotes repetidos inferior (cerca de 30 a 40%) face à estratégia de *Broadcast*.

5.6 Considerações do Capítulo

Analisando todos os resultados obtidos é possível concluir que, quando o alcance WAVE aumenta (neste caso, de 500 para 700 metros), a média de iterações necessárias para entregar os pacotes de emergência diminui. Isto deve-se ao facto de os veículos possuírem mais vizinhos quando têm maior alcance WAVE, fazendo com que os pacotes tenham menos saltos (iteraões).

Quanto à taxa de entrega de pacotes, é visível que as estratégias que apresentam melhores resultados, isto é, maior percentagem de pacotes de emergência entregues são a de Alta Densidade e *Broadcast*. É esperado que este comportamento seja muito semelhante nestas duas estratégias (acontece em todos os cenários testados ao longo das ZoRs). No entanto, a estratégia de *Broadcast* apresenta uma percentagem de pacotes repetidos bastante elevada (cerca de 90%), comparativamente à estratégia de Alta Densidade (cerca de 50%). A estratégia de Baixa Densidade é aquela que tem pior desempenho quer em cenários de alta ou baixa densidade, pois tem uma grande percentagem de pacotes repetidos, tendo em conta a baixa taxa de pacotes entregues. Além disso, como o critério de seleção dos nós retransmissores é baseado na quantidade de vizinhos que cada nó tem atrás, existe a possibilidade de um nó escolhido como *MaxRank* já ter recebido o pacote de emergência de outro *MaxRank* anterior e, por isso, não consegue considerar-se como nó retransmissor. Portanto, a estratégia com melhor desempenho é a de Alta Densidade, uma vez que apresenta uma elevada taxa de entrega de pacotes e uma percentagem baixa de pacotes redundantes, comparando com a estratégia de *Broadcast*.

Em suma, ao analisar os resultados obtidos para as três ZoRs, verifica-se que o número médio de iterações aumenta quando o tamanho da ZoR também aumenta. Este resultado é esperado, uma vez que a extensão da ZoR é maior e, consequentemente o número de veículos também vai aumentando.

Capítulo 6

Conclusões e Trabalho Futuro

6.1 Conclusões

Esta dissertação estudou os mecanismos existentes para disseminar mensagens de emergência numa VANET, implementou estratégias que permitissem utilizar apenas comunicação V2V para comunicar a ocorrência de um acidente num cenário de estrada, para várias densidades de veículos. Para tal, foram implementadas três estratégias (Alta Densidade, Baixa Densidade e *Broadcast*), tendo em conta: utilização apenas de comunicações *V2V*, criação de processos de seleção de nós (veículos) que permitam reduzir a quantidade de pacotes repetidos na rede e tirar partido da localização geográfica de cada nó, delimitando uma área, *Zone of Relevance (ZoR)*, onde a mensagem vai ser disseminada. As duas estratégias de Alta e Baixa Densidade fazem uma seleção de nós, tendo em conta critérios distintos, e foram desenvolvidas para densidades de veículos diferentes (baixa e elevada quantidade de veículos). A terceira (*Broadcast*) não contém qualquer tipo de seleção de nós, nem tem em conta a densidade de veículos, isto é, todos os nós dentro da ZoR enviam a mensagem para todos os seus vizinhos.

A implementação das estratégias de Alta e Baixa Densidade foi baseada em protocolos de *geocast*, uma vez que existe uma área definida onde a mensagem vai ser disseminada (ZoR). A estratégia de Alta Densidade seleciona os nós que vão reenviar a mensagem de emergência (*MaxRanks*). O *MaxRank* é o nó mais distante do nó que enviou previamente a mensagem, mas que ainda está ao alcance do mesmo. O método de seleção da segunda estratégia usa a escolha sucessiva dos nós que tiverem mais vizinhos atrás de si, permitindo atingir o máximo de OBUs com apenas um nó e um salto. A terceira e última estratégia não possui *MaxRanks*, isto é, todos os veículos (OBUs) contidos na ZoR enviam o pacote de emergência para todos os seus vizinhos e, portanto, a quantidade de pacotes repetidos é bastante elevada. Em geral, esta estratégia precisa de mais saltos (iterações) para disseminar o pacote por todas as OBUs que estão dentro da ZoR.

Através dos gráficos obtidos pode concluir-se que a melhor estratégia é a de Alta Densidade, pois é a que apresenta melhores resultados no global, porque apesar de ter uma taxa de entrega de pacotes semelhante à da estratégia de *Broadcast*, a taxa de pacotes redundantes da estratégia de Alta Densidade é bastante inferior à de *Broadcast* (cerca de 40%). Além disso, a quantidade de iterações necessárias para que o pacote chegue a todos os veículos dentro da ZoR é mais elevada na estratégia de *Broadcast* do que na de Alta Densidade. A estratégia de Baixa Densidade mostrou-se ineficiente em todos os casos.

Também é possível concluir que, quanto maior for o alcance WAVE dos veículos, menor

será o número de iterações necessárias, nas estratégias de Alta Densidade e *Broadcast*, pois os *MaxRanks* terão mais vizinhos, logo em cada iteração conseguem chegar a mais veículos. Por fim, variando o comprimento da ZoR, verifica-se que o número de iterações utilizadas aumenta, à medida que o comprimento da ZoR também aumenta.

6.2 Trabalho Futuro

De seguida são apresentadas algumas sugestões para trabalho futuro relacionadas com a disseminação de mensagens de emergência:

- Realização de testes e avaliação do desempenho das três estratégias em cenários de mobilidade;
- Estudo de novas estratégias e testes para cenários urbanos;
- Estudo e avaliação das três estratégias numa rede real;
- Estudo de novas estratégias aplicadas a outras situações de emergência como, por exemplo, uma ultrapassagem ou travagem brusca.

Bibliografia

- [1] Hassnaa Moustafa and Yan Zhang. Chapter 1: Introduction to vehicular networks. In *Vehicular Networks Techniques, Standards, and Applications*, page 1–20. Auerbach Publications, Abril 2009.
- [2] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, pages 27–34, New York, NY, USA, 2003. ACM.
- [3] Intelligent Transportations Systems. Its infographics. [Online] Disponível: <http://www.its.dot.gov/infographs/index.htm>, 2015.
- [4] Intelligent Transport Systems (ITS). Vanets. [Online] Disponível: <http://www.vanet.mdx.ac.uk/>, Janeiro, 2017.
- [5] A. Festag. Cooperative intelligent transport systems standards in europe. *IEEE Communications Magazine*, 52(12):166–172, December 2014.
- [6] D. Jiang and L. Delgrossi. Ieee 802.11p: Towards an international standard for wireless access in vehicular environments. In *VTC Spring 2008 - IEEE Vehicular Technology Conference*, pages 2036–2040, May 2008.
- [7] Sajjad Akbar Mohammad, Asim Rasheed, and Amir Qayyum. Vanet architectures and protocol stacks: a survey. In *International Workshop on Communication Technologies for Vehicles*, pages 95–105. Springer, 2011.
- [8] B. S. Gukhool and S. Cherkaoui. Ieee 802.11p modeling in ns-2. In *2008 33rd IEEE Conference on Local Computer Networks (LCN)*, pages 622–626, Oct 2008.
- [9] Trial-use standard for wireless access in vehicular environments (wave) - resource manager. *IEEE Std 1609.1-2006*, pages 1–71, Oct 2006.
- [10] Microsoft. The osi model's seven layers defined and functions explained. [Online] Disponível: <https://support.microsoft.com/en-us/help/103884/the-osi-model-s-seven-layers-defined-and-functions-explained>, Fevereiro 2017.
- [11] Ieee trial-use standard for wireless access in vehicular environments - security services for applications and management messages. *IEEE Std 1609.2-2006*, pages 0–1–105, 2006.

- [12] Ieee trial-use standard for wireless access in vehicular environments (wave) - networking services. *IEEE Std 1609.3-2007*, pages 1–99, April 2007.
- [13] Ieee trial-use standard for wireless access in vehicular environments (wave) - multi-channel operation. *IEEE Std 1609.4-2006 -Test*, pages 1–82, Nov 2006.
- [14] Felipe Domingos Da Cunha, Azzedine Boukerche, Leandro Villas, Aline Carneiro Viana, and Antonio A. F. Loureiro. Data Communication in VANETs: A Survey, Challenges and Applications. Research Report RR-8498, INRIA Saclay ; INRIA, March 2014.
- [15] Azzedine Boukerche, Horacio A. B. F. Oliveira, Eduardo F. Nakamura, and Antonio A. F. Loureiro. Vehicular ad hoc networks: A new challenge for localization-based systems. *Comput. Commun.*, 31(12):2838–2849, July 2008.
- [16] Bijan Paul, Md. Ibrahim, and Md. Abu Naser Bikas. Article: Vanet routing protocols: Pros and cons. *International Journal of Computer Applications*, 20(3):28–34, April 2011. Full text available.
- [17] R. Kumar and M. Dave. A comparative study of various routing protocols in VANET. *International Journal of Computer Science Issues (IJCSI)*, 8(4), Jul 2011. Full text available.
- [18] J. Kakarla1 S. S. Sathya1 B. G. Laxmi and R. Babu B. A survey on routing protocols and its issues in VANET. *International Journal of Computer Applications*, 28(4), August 2011. Full text available.
- [19] S. Allal and S. Boudjit. Geocast routing protocols for vanets: Survey and guidelines. In *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 323–328, July 2012.
- [20] S. Allal and S. Boudjit. Geocast routing protocols for vanets: Survey and guidelines. In *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 323–328, July 2012.
- [21] I. D. Chakeres and E. M. Belding-Royer. Aodv routing protocol implementation design. In *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings.*, pages 698–703, March 2004.
- [22] Zhang Hequn and Wang Rui. Simulation-based performance comparisons of geocast routing protocols. Master’s thesis, Halmstad University, School of Information Science, Computer and Electrical Engineering (IDE), 2014.
- [23] A. Bachir and A. Benslimane. A multicast protocol in ad hoc networks inter-vehicle geocast. In *The 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring.*, volume 4, pages 2456–2460 vol.4, April 2003.
- [24] A.V. Vasilakos, Y. Zhang, and T. Spyropoulos. *Delay Tolerant Networks: Protocols and Applications*. Wireless Networks and Mobile Communications. CRC Press, 2016.
- [25] W.Richard Stevens Kevin R. Fall. *TCP/IP Illustrated, Volume 1: The Protocols*, 2011.

- [26] A. Hooke L. Torgerson R. Durst K. Scott K. Fall V. Cerf, S. Burleigh and H. Weiss. Delay-tolerant networking architecture,"rfc 4838 (informational), internet engineering task force. [Online] Disponível: <http://www.ietf.org/rfc/rfc4838.txt>, Abril 2007.
- [27] G. Pessoa. Content distribution in vehicular networks using delay-tolerant communication mechanisms. Master's thesis, Universidade de Aveiro, 2015.
- [28] F. Warthman et al. Delay-and disruption-tolerant networks (dtms). *Interplanetary Internet Special Interest Group*, pages 1–35, 2012.
- [29] A. Buchenscheit, F. Schaub, F. Kargl, and M. Weber. A vanet-based emergency vehicle warning system. In *2009 IEEE Vehicular Networking Conference (VNC)*, pages 1–8, Oct 2009.
- [30] B. Ramakrishnan, R. Bhagavath Nishanth, M. Milton Joe, and M. Selvi. Cluster based emergency message broadcasting technique for vehicular ad hoc network. *Wireless Networks*, 23(1):233–248, 2017.
- [31] Hayder Salman Dawood and Yuming Wang. An efficient emergency message broadcasting scheme in vehicular ad hoc networks. *International Journal of Distributed Sensor Networks*, 9(11), 2013.
- [32] A. Pereira and H. Shahnasser. The vehicular environment and its effect on broadcast protocols in vanets. In *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, pages 553–558, Sept 2008.
- [33] M. Durresi, A. Durresi, and L. Barolli. Emergency broadcast protocol for inter-vehicle communications. In *11th International Conference on Parallel and Distributed Systems (ICPADS'05)*, volume 2, pages 402–406, July 2005.
- [34] LUCIANO BONONI, MARCO DI FELICE, and SARA PIZZI. Dba-mac: Dynamic backbone-assisted medium access control protocol for efficient broadcast in vanets. *Journal of Interconnection Networks*, 10(04):321–344, 2009.
- [35] S. Muthu Sundari M. A. Berlin. Cluster based message dissemination for broadcasting the emergency/warning messages using a novel scheduling in vanets. In *International Journal on Computer Science and Engineering (IJCSE)*, Jan 2011.
- [36] Leyli Mohammadkhanli Aylin Deljavan Ghodrati. A new cluster-based efficient broadcast algorithm for alert message dissemination in vanets. In *International Research Journal of Applied and Basic Sciences*, pages 1235–1244, Dec 2013.
- [37] Dongxu Jin, Fei Shi, and JooSeok Song. Cluster based emergency message dissemination scheme for vehicular ad hoc networks. In *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication*, IMCOM '15, pages 2:1–2:8, 2015.
- [38] Liren Zhang and Hesham El-Sayed. A novel cluster-based protocol for topology discovery in vehicular ad hoc network. *Procedia Computer Science*, 10:525 – 534, 2012.

- [39] A. S. K. Mammu, U. Hernandez-Jayo, and N. Sainz. Cluster-based mac in vanets for safety applications. In *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1424–1429, Aug 2013.
- [40] M. and Nishanth R. Bhagavath Ramakrishnan, B. and Selvi. Efficiency measure of routing protocols in vehicular ad hoc network using freeway mobility model. *Wireless Networks*, 23(2):323–333, 2017.
- [41] P. Ganeshkumar and P. Gokulakrishnan. Road accident prevention with instant emergency warning message dissemination in vehicular ad-hoc network. *PLoS ONE* 10(12), pages 1–36, Dezembro 2015.
- [42] P. Ganeshkumar and P. Gokulakrishnan. Emergency situation prediction mechanism: A novel approach for intelligent transportation system using vehicular ad hoc networks. *The Scientific World Journal*, vol. 2015, Article ID 218379, page 12 pages, Abril 2015.
- [43] I. Rubin, Y. Y. Lin, A. Baiocchi, F. Cuomo, and P. Salvo. Vehicular backbone networking protocol for highway broadcasting using directional antennas. In *2013 IEEE Global Communications Conference (GLOBECOM)*, pages 4414–4419, Dec 2013.
- [44] Andrea Baiocchi Francesca Cuomo Pierpaolo Salvo Izhak Rubin, Yu-Yu Lin. Rapid dissemination of public safety message flows in vehicular networks. In *Journal of Communications Vol. 9, No. 8*, Agosto 2014.
- [45] Francesca Cuomo, Izhak Rubin, Andrea Baiocchi, and Pierpaolo Salvo. Enhanced {VANET} broadcast throughput capacity via a dynamic backbone architecture. *Ad Hoc Networks*, 21:42 – 59, 2014.
- [46] I. Rubin, A. Horng, and C. Y. Yang. Lane based backbone synthesis protocols for vehicular ad hoc networks. In *2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, pages 95–102, June 2014.
- [47] S. M. Tornell, C. T. Calafate, J. C. Cano, and P. Manzoni. Dtn protocols for vehicular networks: An application oriented overview. *IEEE Communications Surveys Tutorials*, 17(2):868–887, Secondquarter 2015.
- [48] iMatix Corporation. Zmq. [Online] Disponível: <http://zeromq.org/>, Junho, 2016.
- [49] G. Pessoa, R. Dias, T. Condeixa, J. Azevedo, L. Guardalben, and S. Sargento. New insights on content distribution emulation for vehicular networks. In *The Wireless Days Conference 2017*, March 2017.
- [50] Matlab. Mathworks. [Online] Disponível: <http://www.mathworks.com/products/matlab/>, 2016.
- [51] Jeremy J. Blum and Azim Eskandarian. A reliable link-layer protocol for robust and scalable intervehicle communications. 8(1):4–13, March 2007.
- [52] Muhammad Awais Javed, Duy Trong Ngo, and Jamil Yusuf Khan. A multi-hop broadcast protocol design for emergency warning notification in highway vanets. 2014(1):179, 2014.